AFRL-SN-RS-TR-2001-146 Vol I (of VI)
Final Technical Report
July 2001

KNOWLEDGE BASE APPLICATIONS TO ADAPTIVE SPACE-TIME PROCESSING, VOLUME I: SUMMARY

ITT Systems

Yassar Salama and Roy Senn

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-SN-RS-TR-2001-146 Vol I (of VI) has been reviewed and is approved for publication.

APPROVED: Michael C Wicks

MICHAEL C. WICKS
Project Engineer

FOR THE DIRECTOR: Robert G. Polce

ROBERT G. POLCE, Chief
Rome Operations Office
Sensors Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/SNRT, 26 Electronic Pky, Rome, NY 13441-4514. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	JULY 2001	Final Aug 95 - Jul 99

4. TITLE AND SUBTITLE KNOWLEDGE BASE APPLICATIONS TO ADAPTIVE SPACE-TIME PROCESSING, VOLUME I: SUMMARY	5. FUNDING NUMBERS C - F30602-95-C-0041 PE - 62702F PR - 4506 TA - 11 WU - 2B
6. AUTHOR(S) Yassar Salama and Roy Senn	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) ITT Systems 775 Daedalian Drive Rome New York 13441	8. PERFORMING ORGANIZATION REPORT NUMBER N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/SNRT 26 Electronic Pky Rome New York 13441-4514	10. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-SN-RS-TR-2001-146 Vol I (of VI)

11. SUPPLEMENTARY NOTES

Air Force Research Laboratory Project Engineer: Michael C. Wicks/SNRT/(315) 330-2556

12a. DISTRIBUTION AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.	

13. ABSTRACT *(Maximum 200 words)*

This report summarizes a three-year development of an innovative concept for coordinated adaptive radar processing using knowledge-based control. Under this approach, knowledge and expert rules about the operating theater are used to select algorithms and training data, thereby significantly improving the performance of modern adaptive array radars in dynamic and non-homogeneous environments.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Radar, Algorithms, Signal Processing, Space-Time Adaptive Processing, STAP	72
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

Table of Contents – Cont'd

Table of Contents – Cont'd

List of Figures

List of Tables

(This page intentionally left blank)

1.0 Purpose

This Final Report documents the technical milestones and findings achieved in the performance of the Air Force Research Laboratory AFRL/SNRT (Rome, NY) Contract No. F30602-95-C-0041. This effort titled "Knowledge Base Applications to Adaptive Space-Time Processing" reports on the three-year development of an innovative concept for coordinated adaptive radar processing using knowledge-based control. Under this approach, knowledge and expert rules about the operating theater are used to select algorithms and training data thereby significantly improving the performance of modern adaptive array radars in dynamic and non-homogeneous environments.

This document is intended to give the reader a comprehensive perspective on the scope and accomplishments of the program but does not necessarily give a fully detailed picture of the extent of the effort. The reader should start here but, for a closer look at the theoretical development, implementation and experimentation that was conducted, the references should be consulted.

2.0 Background

The intelligent control of surveillance radar systems has long been recognized for its potential. In 1986 the Resource Allocator program varied system and waveform parameters based on assessment of the radar environment. For example, if a jammer were determined to be located in a certain direction, a radar system could increase the transmitted power in that direction until the required burn through power is reached, form a null in the jammer direction, or avoid the problem sector. In 1989 knowledge-based control was used to demonstrate improved tracking performance [1]. This Knowledge Based Tracker (KBT) program demonstrated improved tracking performance through the use of external knowledge sources. In 1991, artificial intelligence techniques were introduced into the detection stage of the radar processor [2]. Significant detection performance improvements were demonstrated by careful selection of constant false alarm rate (CFAR) algorithms and their parameters, based upon the observed environment.

The AFRL contract for this reported research was put in place in August 1995. The contract continued previous work in the knowledge based tracker and detection stages, and extended that work to include the filtering stage of the radar signal processor. In traditional space-time adaptive processing (STAP), the filter forms an estimate of the interference in the range cell under test through the use of secondary or training data. Since the covariance matrix of the interference is only estimated and not known exactly, cancellation is often far from optimum. STAP performance is therefore limited by how well the training data represents the interference in the cell under test.

In the approach presented in this report, various knowledge sources that can provide information about what is observed and the radar environment are used to select the training data, STAP algorithms, and parameters. This knowledge-based control (KBC) also helps to direct pre-adaptive processing, which may reduce the burden on the STAP processor and lower the required degrees of freedom (DOF). For example, mapping data may indicate the presence of known discretes, ground traffic, or other interference sources. A KBC approach could then direct deterministic nulling of the interference in the spatial or temporal domains prior to adaptivity. A knowledge-based KBSTAP would then be able to handle the underlying clutter distribution with fewer DOF.

A knowledge-based approach to radar processing requires an integrated approach to the radar operating environment and a team with multi-disciplinary radar skills. The development team consisted of researchers who had a track record in the technology areas required by a knowledge-based approach. The prime contractor, ITT Systems (formerly Kaman Sciences Corporation), had developed a highly successful Expert System (ES) CFAR processor with team member Syracuse University. ITT Systems also developed, with team member Technology Service Corporation (TSC), the Research Laboratory Space-Time Adaptive Processing (RLSTAP) tool, which enables researches to assemble radar data and algorithms in a visualization environment.

Other team members had a record of performance on related AFRL radar programs. SRC had performed numerous studies in the area of STAP. Technology Service Corporation (TSC) co-developed RLSTAP, had a long history of bistatic radar analysis, and was developing a KBT. Decision-Science Application (DSA) developed the Bistatic Radar Simulation (BRADS), which computes bistatic clutter levels and system performance for bistatic concept development and evaluation. Finally, Capraro Technologies Inc. was a key motivating force behind the ES-CFAR concept.

3.0 Objectives

The overall objectives of this reported effort were to:

- Develop and evaluate signal processing algorithms for monostatic radar,
- Exploit knowledge-based system and other state-of-the-art software solutions to improve system performance in a dynamically changing radar environment.

3.1 Signal Processing Algorithms

Algorithm development was to encompass the four fundamental stages of surveillance radar: data generation, filtering, detection, and tracking, as investigation warranted.

3.1.1 Data Generation

Data generation would provide known inputs to the KBSTAP for testing, validation, and evaluation. Physical models, representative models, and measured data were to be investigated as alternate or complementary approaches to data generation. Physical models simulate radar returns from environments based on specific scenario geometries and sensor parameters. Representative models provide control over statistical properties of simulated space-time data, as calculated from parameters characterizing the phased-array radar platform, environmental clutter, signals, and jammers. Measured data, collected in the field under controlled conditions, provide the best test of signal processing algorithms operating in a realistic environment and includes error sources that may not be accounted for in simulated data.

3.1.2 Filtering

The filter process performs spatial and temporal filtering of the received data. This process includes analog and digital beam formers, space/space-range adaptive processing, and adaptive and non-adaptive space-time processing. KBSTAP was to be further developed under this effort to provide maximum flexibility with knowledge based decisions in the configuration of spatial and temporal filters.

3.1.3 Detection

Automatic detection schemes declare and record target detections from received signals, without human interpretation and intervention. Target detections are declared when a signal exceeds a specified threshold level that may be determined by a number of factors such as signal-to-noise ratio, probability of detection, probability of false alarm, and the statistics of the target and background. Adaptive threshold techniques are employed to control false alarm rates in varying background environments. The most common of these is a processor that is designed to maintain a constant false alarm rate (CFAR) by adjusting the threshold for an individual sample or cell by estimating the interference in the neighborhood of the cell. Since the effects of the radar environment are likely to differ through each stage, the detection process must also be able to select algorithms, parameters, and reference samples based on knowledge of the environment.

3.1.4 Tracking

The knowledge base tracking process receives detections from the detection process and target, clutter, and jammer information through a tracker environmental processor, which helps it to select track filter algorithms and gate windows. The tracking process then performs track initiation/drop, state promotion, prediction, maneuver detection, and other functions to maintain and report identified tracks (See Volumes IV and V).

3.2 Software Solutions

Exploitation of state of the art software solutions was to be derived from knowledge based engineering techniques/concepts and the establishment of KBSTAP software requirements. Knowledge-based systems are recognized as having value both in the dynamic selection or tailoring of appropriate radar algorithms, and in the design of software tools that would aid the maintenance of KBSTAP. Major elements of a knowledge based architecture that were to be considered included:

- <u>A user interface component</u> that would consist of the basic man-machine interface for signal processing functionality. This component would provide the underlying functionality for processing line-up and control parameter specification.

- A data visualization component that would encompass the graphical data presentation functionality. The basic capability would include 2-D and 3-D data presentation with some level of end-user programmable, graphic-display functionality.

- A knowledge base component that would encompass all of the knowledge base related functions of the system, for example rule maintenance and selection and knowledge maintenance.

- A data management component that would include all data interfaces, data transports, data translation, data formatting, and data storage functionality. The component would need to support existing data formats and multiple representative methods.

- A process management component that would supply the basic functionality related to process invocation, process scheduling and inter-process communication, including both data and control parameters.

- A process builder component that would include all of the software engineering tool functionality for describing, building and maintaining the signal processing software. This component would include the tools, conventions, and interfaces that allow inclusion of traditional programming language as well as specialized application command languages.

The objectives for KBSTAP software were extensive and encompassed consideration of:

- Support of data types sufficient to represent and process radar data and algorithms was required. Selection of the most compact type providing the greatest dynamic range would be important when handling large data sets typically produced by radar sensors. Processing time and system resources would be additional factors in determining the best representation.

- Support for databases was to include ongoing programs such as RLSTAP/ADP, synthetic data sets, MATLAB, formatted data sets, DTED/DFAD data sets, and data sets that evolve from future programs and sources of knowledge.

- Programming language capability was to include the compilation and library support for languages such as MATLAB, C, C++, shell script, and assembly. Library and linking functions needed to support cross platform operations such as rebuild and reuse code between an IBM AIX machine and a SUN SPARC.

- A graphic user interface (GUI) and visualization services were to provide a windows view of all the functions necessary to create, interactively change, and maintain image display and manipulation, color map control, 2D and 3D plotting, surface rendering, annotations, and animations to allow visualization of complex radar models. Programming via a visual language expressed as data flow diagrams such as that available to RLSTAP was required to allow non-programmers to apply sophisticated algorithms and programming concepts.

- A simulation environment that was programmable, flexible and, extendible was required. Capabilities were to include:

- Rapid system prototyping that enables the incremental integration and cost effective evolution of software systems.

- Distributed simulation and synthetic environments that can integrate real as well as virtual objects, in terms of both their visual and computational descriptions, for the creation of synthetic worlds.

- Software libraries and structures that support the development of common architectures and interfaces, increase the potential for reusability across underlying models of computation, a diversity of programming languages, and varying degree of assurance.

- Non-proprietary and COTS software was to be used to best advantages so the end user would not be overburdened with the purchase, maintenance and development of specialized requirements.

- Unique functional requirements such as feedback loops, needed to be addressed to understand how an integrated environment would accommodate them. Having open interface design that allow other systems to be integrated easily was seen to be a necessity.

- Leverage from previous investments was to be maximized. A significant experience base and software environment would be leveraged from the AFRL RLSTAP/ADT program. RLSTAP/ADT is a complete user-friendly radar environment consisting of simulated and measured data, a full library of signal processing routines, and a flexible graphical interface.

- User and development support was required. Tools and methodologies were to be put in place for maintenance, enhancement, and development. Documentation, distribution, and on line assistance were to be provided.

- The user base was to be expanded and exploited to allow open collaboration resulting in new techniques, product direction consortiums, end user feedback, and distribution mechanisms.

- Parallel processing opportunities were to be considered for distribution of tasks and a range of granularity to overcome limitations of CPU cycles, I/O bandwidth, or memory. The STAP environment suggests a natural transition to a parallel process.

4.0 Milestones

This section documents evolution of thought, and record of achievement that occurred over the duration of the contract. Key elements of the effort were a coordinated development of a knowledge-based radar architecture, and research by specialists who understood that their contributions would need to function within that architecture, and could also take advantage of it. For example, both filtering and tracking algorithms could utilize the same knowledge sources provided by a central Knowledge Base Controller (KBC) to remove interference and maintain tracks through interference. The organization of the following subsections follows specific activities where substantial effort was invested.

4.1 Project Startup

To start the KBSTAP project off strongly, a kickoff meeting was held on 12 September 1995. A second meeting on 29 September 1995 quickly followed this. The kickoff meeting introduced team members and reviewed the proposed objectives, history, and previous algorithm and software development within a knowledge-based environment. Also presented were a project schedule and a proposed Design Plan (See the Section on the Design Plan below.).

The follow-up meeting focused on the application of various filtering algorithms and knowledge-based control of STAP. A generalized STAP configuration was presented, which was discussed as a baseline for KBSTAP. Known filtering techniques were also discussed, along with their requirements and principal effects. Table 1 lists those filters that were covered along with their application. The performance of each of these filters were illustrated with elucidating freehand sketches of the Doppler effects expected. An example of these sketches is shown in Figure 1.

TECHNIQUE	REQUIREMENTS	APPLICATIONS
DPCA	• Slow Moving Tx	• Rx Mainlobe Clutter
Clutter Tuning	• Confined Target region • Specialized geometry	• Tx and Rx Mainlobe Clutter
Pattern Synthesis	• Interference Directions	• Mild Jamming • Clutter Discretes
Adaptive DPCA	• Slow Moving Tx • Large Training Data Set	• Rx Mainlobe Clutter • Mild Jamming
Displaced (or Adjacent) Beam	• Confined Interference • Small Training data Set	• Rx Mainlobe Clutter • Mild Jamming
Pre-Doppler Two-Step	• Confined Clutter • Small Training data Set	• Rx Mainlobe Clutter • Strong Jamming
Doppler Filter/AP	• Long CPI	• Clutter • Mild Jamming
PRI (pulse repetition interval) Staggered	• Large Training data Set	• Clutter • Mild jamming
Adjacent Filter	• Large Training data Set	• Rx Mainlobe Clutter • Mild Jamming
Joint Domain	• Confined Interference • Small Training Data Set	• Rx Mainlobe Clutter • Mild Jamming
Post Doppler Two-Step	• Confined Clutter	• Rx Mainlobe Clutter • Strong Jamming

Table 1: Filter Characteristics

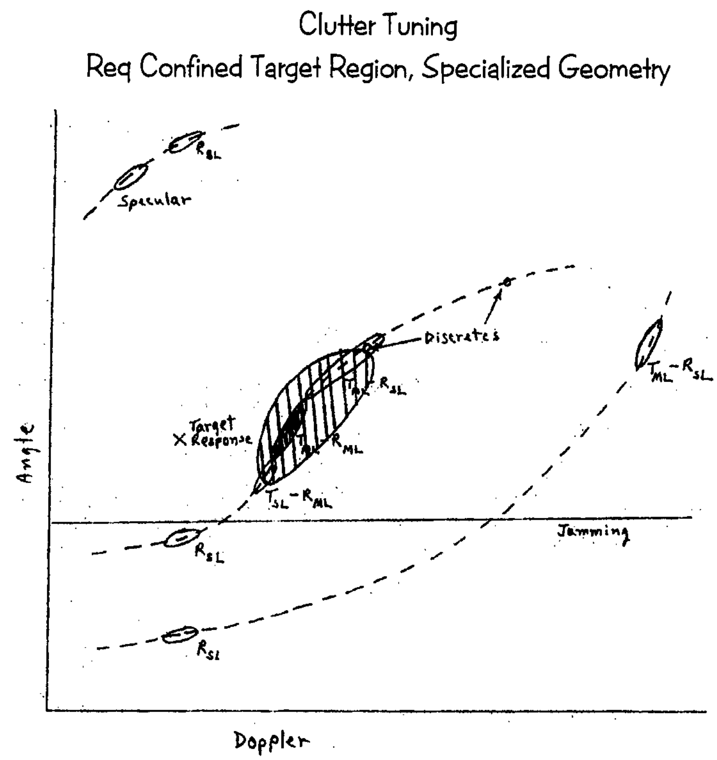


Figure 1: Sketch of Filter Doppler Effects

4.2 Design Plan

A Design Plan (DP) [3] was proposed as an essential first step toward the development of a knowledge-based STAP environment. A well thought out DP was expected to save countless hours as the project progressed through design and implementation. Specifically the DP was viewed as a vehicle for:

- Identifying the unique and essential components necessary for monostatic/bistatic sensor development, analysis and evaluation
- Defining the equations, definitions, models, and procedures for space-time processing (STP)
- Defining the equations, definitions, models, and procedures for monostatic/bistatic, clutter modeling and analysis, and the development of an environmental processor
- Defining the equations, definitions, models, and procedures for integrating knowledge base engineering techniques with STP detection and tracking schemes, to include the architecture of the knowledge base processor, which incorporates such considerations as, but not limited to, parallel and real-time processing.
- Defining the procedures for validating and benchmarking the various algorithm developments

- Defining the equations, definitions, models, and procedures for synthesizing target, clutter, and jamming data

- Identifying all software requirements, including data formats, supported databases and programming languages, and modular and programmable features.

- Including provisions for analyzing measured data as it becomes available

The DP was seen to be crucial to this effort's success because of the many factors and constraints that needed to be considered in an ordered and coordinated fashion. The initial KBSTAP architecture presented in the DP is shown in Figure 2. The entire signal processor chain is under the direction of a Knowledge Base Controller (KBC). The KBC would receive external data regarding the radar configuration, clutter data, flight profiles, intelligence data, and information, from other systems. This information would be used to select programmed configurations of the processing chain, select algorithms and their parameters for execution, and select and condition secondary data for the filtering, detection, and tracking processes.

Beam forming could be implemented in a number of ways, including an analog beamformer, a digital beamformer, or an adaptive space-time filter. A space/space-range processor could also be implemented to cancel interference caused by jammers or terrain scattered interference. At the radar front end, the KBC would pass information regarding the environment to the space/space-range processor and digital combiners. This information might include jammer or clutter range, Doppler, spread, and location. This would allow the patterns to be formed dynamically based on the assessed environment. The outputs of the space/space-range processor and digital combiner would in turn be fed back to the KBC to report on how well the interference is suppressed. The pattern formed by the space/space-range processor would be subtracted from the main array pattern to reduce the number of DOF required in the adaptive processor.

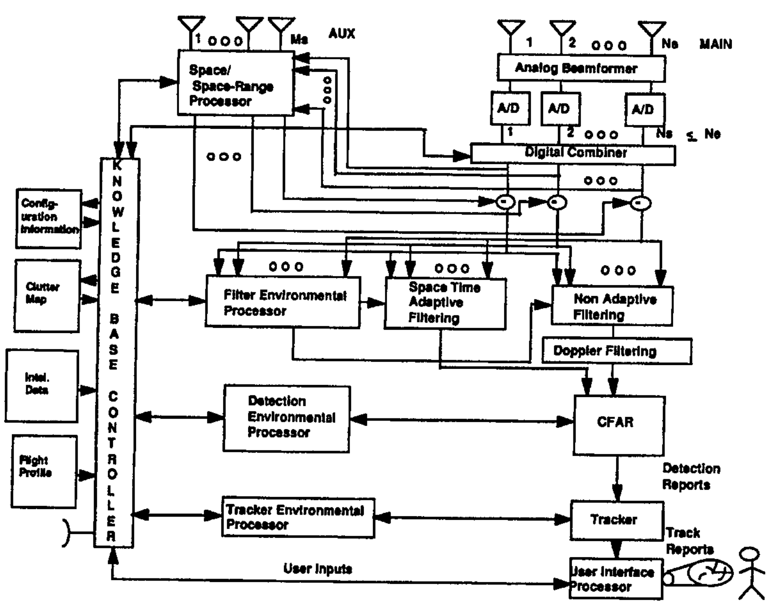


Figure 2: Initial KBSTAP Architecture

The filtering stage could be implemented using adaptive or non-adaptive techniques, or a combination of both, depending on the environment. A known key advantage of non-adaptive approaches was that an estimate of the covariance matrix of the environment is not needed. In the adaptive approaches, selecting reference cells that are representative of the test cell would be an issue. The estimate of the covariance matrix could only be as good as the samples supporting the estimate. A fundamental objective of the KBSTAP effort was to improve adaptive processing performance by careful selection and conditioning of the reference cells.

In the filtering, detection, and tracking stages, the KBC would interact with the environmental processes to select and condition reference samples, to select algorithms, and to select parameters for the chosen algorithm. The interaction would be two-way, since the results of each stage may impact the control of other stages. For example, adaptive filtering may result in a certain amount of whitening of the clutter. That information may impact the selection of a CFAR algorithm to operate on the output data. Ideally, the output of adaptive filtering process would be fully whitened, resulting in the selection of cell-averaging CFAR. In practice, the output would not be completely whitened, requiring the intelligent selection of the CFAR processor, including selection of algorithm parameters, and windows. In addition, one stage may alter the environmental parameters of subsequent stages. A space/space-range processor, for example, may eliminate a jammer prior to the detection process, so that the detection environmental processor does not need to pass jammer information to the detector. Therefore, feedback from each stage would become an important knowledge source.

An additional benefit of overall knowledge base control would be that coupling between each stage of the process can be accounted for. For example, some STAP algorithms include an embedded CFAR capability. In those cases, the KBC would short-circuit the detection stage. Also, the knowledge based tracker is dependent on the setting of the threshold for detection. The tracker may require that the threshold be lowered to allow more detections to be placed into the track. It is therefore important for the KBC to provide inputs to each stage, monitor the performance measures of the outputs, and accept feedback in the form of modified description of the environment, performance measures at each stage, and recommendations where appropriate.

The DP [3] laid out a development plan for each of the functional elements, showing how each of the elements relates to others within the KBSTAP architecture. Theoretical descriptions were provided along with essential equations. Comprehensive references were included to provide theoretical and practical background on the particulars of the STAP functional elements.

The DP included a Software Implementation Plan (SIP) delivered to AFRL under a separate cover [4]. The SIP provided a basis for continuing discussion between AFRL and the development team and presented an initial concept of the KBSTAP software architecture from three perspectives:

- An application perspective that described the system internals of data processing
- An operational perspective that described the system in terms of what the user controls
- A structural perspective that described the system in terms of the organization of software infrastructure components.

The SIP also discussed the specifics of the KBSTAP implementation, including the planned operational environment, development environment, development activities, and schedule of milestones. The Design Plan and Software Implementation Plan were delivered to AFRL on 15 April 1996.

4.3 Knowledge-Based STAP

As suggested by the DP [3] the knowledge-based space-time adaptive processor (KBSTAP) can be conveniently viewed in terms of four complementary processes: the KBC, filter processor, detection processor, and track processor (see figure 2). As the project progressed this view provided a means to divide and conquer the technical and theoretical aspects unique to each subsystem while keeping in mind that the original premise of knowledge-based radar processing was that these components would each benefit from shared external and internally discovered knowledge. The following subsections discuss the activities and accomplishments associated with the development of each subsystem.

4.3.1 Knowledge-Based Controller

It was recognized by the DP [3] that KBSTAP would require a component to oversee the overall knowledge-based processing environment. The KBC would use a variety of information to coordinate processing within the filtering, detection, and tracking stages. It would manage platform and radar configuration data, and provides this data to the necessary components. The controller also would accept inputs from mapping sources, external sensors, and other stages, such as the tracker. Using this information, it would populate a database of targets and interference, calculate and maintain priorities and confidence levels, and correlate tracker outputs with the target and interference databases.

When deriving interference information from mapping data, only major interference sources would be considered in order to make the computational burden manageable. These interference sources include: primary highways; major clutter interfaces, such as land/sea or mountains/plains; and significant discretes, such as bridges

Other sources could be added to the interference database if identified by the tracker or other knowledge sources. In addition, the priority of interference derived from mapping data would be elevated if identified through the detection stage, or more significantly, if placed into a track.

The overall approach to KBSTAP was refined and presented at a 06 May 1997 Technical Interchange Meeting (TIM). This restatement was based on the evolution of thought and experimentation that had taken place since the inception of the project, and was intended to establish a consistent framework for further KBSTAP development. Table 2 summarizes these refinements according to the principal elements of KBSTAP. Besides managing the filter, detector, and tracker sub-components, the knowledge-based controller required access to external data (knowledge) sources. These sources, also listed in Table 2, would provide the knowledge necessary for the controller to decide processing steps most appropriate for the radar environment at hand. A standard for Doppler radar data organization that would be utilized for mapping these data sources to the angle/doppler/range space is also illustrated in Table 2.

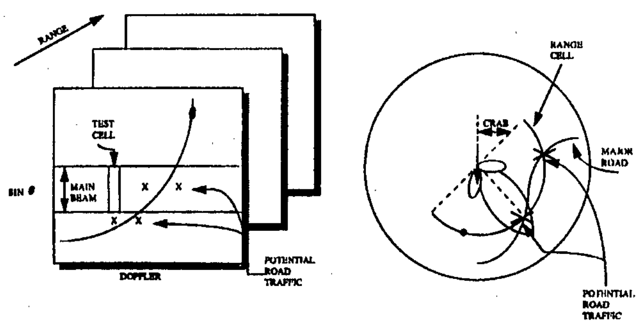
KBSTAP ELEMENTS	APPROACH
Knowledge Based Controller	• Purpose: ▪ Manages Knowledge Sources And Data ▪ Manages Interface Between Radar Processing Stages • Knowledge Sources: Radar Data, Mapping Data, and Reports
Knowledge Sources	• Radar Data ▪ Detections/False Alarms ▪ Statistical Information • Mapping Data: Focuses On Primary Non-Homogeneities: ▪ Road Traffic ▪ Major Discretes, ex. Bridges ▪ Major Clutter Interfaces (Land/Sea; Mountains/Plains) ▪ Shadowing ▪ Obstacles • Reports: AWACS, JSTARS, JSS, Rivet Joint, and Tracker
Mapping to Angle /Doppler /Range Space	
Filtering Stage	• Deterministic Nulling ▪ Spatial & Temporal Nulls Used To Eliminate Moving Interference • Algorithm Selection ▪ Adaptive & Non-Adaptive Algorithms Selected Under Knowledge-Based Control • Sample Selection ▪ Mapping Data Used To Select Coarse Secondary Data Set ▪ Non-Homogeneity Detector Refines Selection • Filtering: Selected Algorithms are Executed
Detection Stage	• Detection Stage Consists Of 3 Components: ▪ CFAR Segmentation – 2-D Segmentation Based On Existing ES-CFAR – Segmentation Augmented By Mapping Data Information ▪ Threshold Computation – 2-D Threshold Multipliers Computed Via Closed Form Solution ▪ CFAR Execution – After Knowledge-Based Filtering & CFAR Segmentation, It Is Anticipated That Cell-Averaging CFAR Will Be Selected In Vast Majority Of Cases • Separate Detection Stage May Not Be Required, Or May Be Very Limited, Depending On Form Of Filtering Performed
Tracking Stage	• Post-Detection Processor ▪ Centroiding ▪ Ambiguity Removal ▪ Data Association • Proactive Track Processor ▪ Anticipates Target Maneuvers Due To Obstacles ▪ Coasts Tracks In Shadowed Regions ▪ Identifies Possible Ground Moving Targets And Discretes

Table 2: Summary of KBSTAP Approach

4.3.2 Filtering Processor

The following subsections describe the actions taken to test, refine and implement the concepts presented in the DP [3]. The primary questions to be answered were the degree to which external (outside sources such as terrain data) or internal (feedback from one or more of the KBSTAP processes) information could be used to improve filtering and what working rules might be suggested to use this available knowledge most effectively.

4.3.2.1 STAP Clutter Simulations

To obtain a basis for creating an illustrative example of knowledge-based decision making at the filtering level, STAP simulations were exercised for antenna and platform geometries representative of Multi-Channel Airborne Radar Measurement (MCARM). Homogeneous and non-homogeneous simulated clutter were included in the simulations. An adaptive DPCA STAP algorithm that employed 11 columns and two pulses (21 DOF) was exercised. Table 3 shows the antenna, platform and waveform configuration and the clutter and filter combinations that were simulated.

It was evident from these simulations that the non-homogeneous clutter, including the insertion of strong discrete scattering, had little detrimental effect on performance. The results, illustrated by computer generated data plots, were briefed to AFRL on 6 March 1996. The consensus was that the sensitivity to clutter non-homogeneity might be far greater when considering actual measured data. Consequently, procedures were established for obtaining actual MCARM data for future simulations.

STAP CLUTTER SIMULATIONS
Antenna:
16 Columns of 8 Elements Each; One receiver per column (16 receivers); Tilt $\theta_t = -5°$
Platform:
• Velocity $V_a = 155.75$ m/s
Target:
• Broadside $\theta_s = 0°$
• Radial Velocity $V_x = 23$ m/s

Clutter	Filter (Monostatic)
Homogeneous • No Errors • Only Non-Ideal Condition "2 $V_a d \neq$ PRI" ($\beta \neq 1$)	Single Column Beamformer (16 Column) Two-Pulse Non-Adapted MTI Two-Pulse Adapted MTI • DOF = 32 • $N_{Samples} = 32$
Non-Homogeneous • No-Errors	Single Column Beamformer (16 Column) Two-Pulse Non-Adapted MTI Two-Pulse Adapted MTI • DOF = 32 • $N_{Samples} = 32$

Non-Homogeneous • 10°RMS Phase Errors	Single Column Beamformer (16 Column) Two-Pulse Non-Adapted MTI Two-Pulse Adapted MTI • DOF = 32 • $N_{Samples} = 32$
Non-Homogeneous • 10°RMS Phase Errors • Pattern Errors	Beamformer (16 Column) Two-Pulse Non-Adapted MTI Two-Pulse Adapted MTI
Homogeneous • C/N = 90 dB	Two-Pulse Adapted MTI • $\theta_e = 10°$ • DOF = 32 • $N_{Samples} = 64$ Two-Pulse Adapted MTI • $\theta_e = 10°$ • DOF = 32 $N_{Samples} = 32$ Two-Pulse Adapted MTI • $\theta_e = 0°$ • DOF = 32 $N_{Samples} = 32$

Table 3: Clutter/Filter Simulations

4.3.2.2 Application of Space-Time Filtering Algorithms to MCARM Monostatic Radar Data

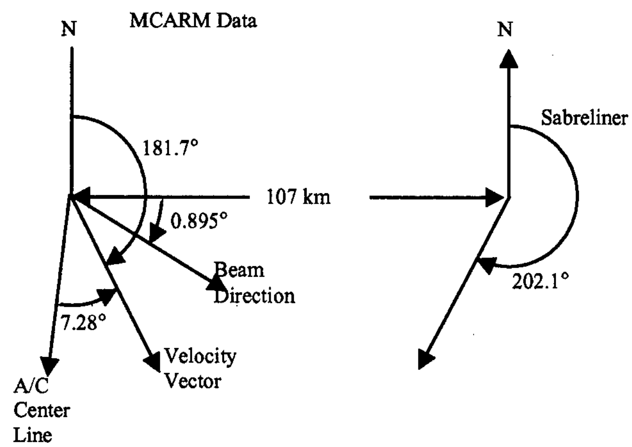
The actual MCARM monostatic radar data obtained as a consequence the clutter simulations made it possible to test a number of space-time filtering algorithms. The flight configuration from this data is shown in Figure 3. Non-adaptive algorithms included:

- Doppler filtering
- Element Space Displaced Phase Center Antenna (DPCA)

STAP algorithms included:

- Factored STAP
- Joint Domain Localized
- Element Space Pre-Doppler ADPCA
- Element Space Post Doppler ADPCA (PRI Staggered)
- Beam Space Pre-Doppler ADPCA
- Beam Space Post-Doppler

The adaptive methods worked reasonably well in that an injected target at about the noise level, as measured at a column, was reasonably well detected. Non-adaptive DPCA (Element Space) did not work as well, probably because of a "mismatch" between the aircraft velocity and the pulse repetition interval (PRI) and antenna errors. Here the sum and difference beams are combined to approximate the "DPCA" condition, at least in the vicinity of the main beam peak, and an adjustment ("gain factor") can be conveniently applied to compensate for the velocity-PRI mismatch.



MCARM Flight 5, Acquisition 575
- Ground Speed = 100.15 m/s
- Drift (Crab) Angle = -7.28°
- Beam Direction = 0.895
- Beam Direction Clutter velocity = 14.24 m/s
- Horizon = 2.28 km (10,000 ft. Altitude)
- Ambiguity Range = 76 km
- Clutter Velocity = 100.6 m/s
- Ambiguity Doppler = 480.0 m/s

Injected Target
- Range Bin 350
- Radial velocity = 20 m/s

Figure 3: MCARM Monostatic Radar Data

Some observations on the performance of the STAP methods were:

1. Element space methods that involved a relatively large number of DOF (N_{DOF}), such as N_{DOF} = 22, worked best with reference cells obtained from a "sliding window" of $N_R = N_{DOF}/2$ and between 1 and 10 dB of diagonal loading (P_n). Beam space methods that relied on relatively fewer DOF, such as $N_{DOF} = 4$, worked well with $N_R = 2N_{DOF}$ for pre-Doppler STAP and $N_R = 5N_{DOF}$ for post-Doppler STAP (P_n = 10 dB). No consistent rule for selecting N_R was evident.

2. Beam Space ADPCA with the well formed MCARM analog sum and difference beams, and two additional sum beams (formed digitally) (N_{DOF} = 4 x 2 = 8) appeared to substantially suppress all clutter and suggested the presence of ground traffic in the vicinity of a major route (Route 13 in Delaware). Element Space ADPCA also demonstrated substantial clutter suppression and yielded strong returns at the Doppler and ranges corresponding to Route 13 traffic.

3. The post-Doppler STAP methods, such as PRI-Staggered, did not suppress main lobe clutter as well as the pre-Doppler STAP methods. This occurrence could have been, perhaps, a consequence of intrinsic clutter motion and/or nonlinear platform motion over a coherent processing interval (CPI). The pre-Doppler STAP methods "track" these motions because adaptive cancellation is reinitiated and applied to every pair of successive pulses. These findings were presented to AFRL on 25 April 1996. Table 4 is a summary of these STAP examples.

STAP Method	N_{DOF}	N_R	P_n(dB)
Factored STAP	22	10	1
ADPCA	22	8	10
PRI Staggered	22	8	10
Beam Space ADPCA (Analog Beams)	4	8	0
Beam Space PRI Staggered (Analog Beams)	4	20	10
Beam Space PRI Staggered (Analog + Digital Beams)	8	6	10
Joint Domain Localized	6	6	1

Table 4: Summary of STAP Examples

4.3.2.3 Application of Six STAP Methods to Measured MCARM Data

Additional observations and examples regarding the application of the six STAP methods to the measured MCARM data (Flight 5, Acquisition 575) were presented at a 21 May 1996 TIM. The beam space methods were applied with combinations of analog beams and digital beams. A "sliding window" reference data selection method was applied. The number of reference cells, N_R, and diagonal loading was varied in each simulation and only the result corresponding to the best performance in the vicinity of the injected target was recorded. Conclusions drawn from these results were:

1. The covariance matrices corresponding to small N_{DOF} methods (beam space methods) can be estimated more accurately than for the large N_{DOF} methods (element space methods) when restricted to selecting range cells for reference data by the "sliding window" method. Usually, $N_{DOF}>N_R$ for the larger N_{DOF} methods. It was presumed that the adaptive filter suppresses more clutter with less target gain loss for the small N_{DOF} methods. Recent work at AFRL on advanced methods of selecting reference data based upon statistical measures has demonstrated improved filtering with larger N_{DOF}. There appeared to be considerable merit to investigating this avenue further.

2. Post-Doppler ADPCA methods are, perhaps, least sensitive to PRI/velocity mismatch or antenna errors (spatial channel mismatch). With ADPCA, such mismatch errors are adaptively compensated for, resulting in a DPCA-like clutter suppression filter. The advantage of preceding the adaptivity with Doppler processing is that the angular extent of the clutter ridge is then limited to that defined by the Doppler filter mainlobe. The limited angular extent ensures the existence of a complex weight vector that can reasonably compensate for antenna channel mismatch and PRI/velocity mismatch. It was noted that antenna channel mismatch is generally angle dependent (complex antenna pattern shapes differ between elements). Also, PRI/velocity mismatch is a time delay error implying that a good correction with complex weights is possible only over a limited angular extent.

3. For small N_{DOF}, such as beam space ADPCA, post-Doppler methods appear superior to pre-Doppler methods. This observation can be explained as follows:

   Because with beam space the number of DOF is few, adaptivity can not totally compensate for all errors (velocity/PRI mismatch, non-uniform motion, antenna mismatch, etc.) in attempting to realize a DPCA-like suppression filter. The adaptivity must be applied as efficiently as possible. With pre-Doppler ADPCA, some clutter that is suppressed adaptively would only have been filtered anyway in the subsequent Doppler processor. However, with post-Doppler ADPCA, only that clutter that passes through the Doppler processor (e.g., Doppler mainlobe clutter) would require suppression adaptively. Post-Doppler ADPCA is therefore superior to pre-Doppler ADPCA when the number of DOF are limited.

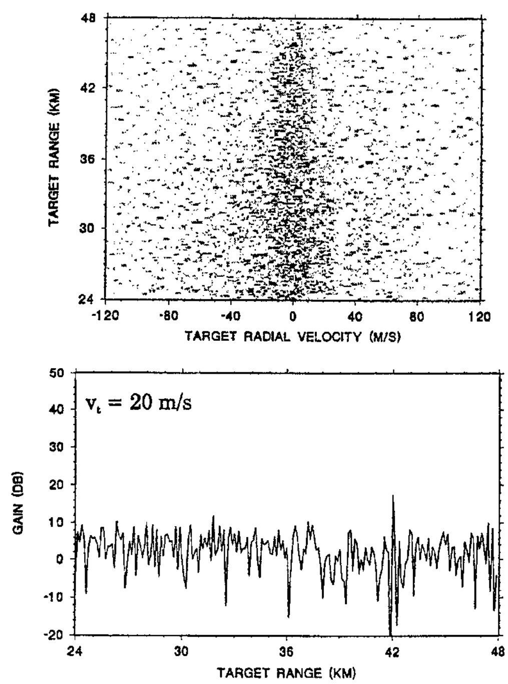
4. Large N_{DOF} methods, such as element space ADPCA, appear to be more sensitive to mainlobe targets if the adaptivity precedes Doppler filtering. What appeared to be ground traffic on a major highway was most readily observed with element space pre-Doppler ADPCA. One proposed explanation was that this method is least sensitive to errors arising from non-uniform platform motion as well as intrinsic clutter motion. In one implementation of this method, successive pairs of PRIs from all the elements (or column subarrays) are applied in the adaptive weight computation. Because a new set of weights is determined for each successive pair of PRIs, the weights approximately "track" phase changes due to non-uniform platform motion and intrinsic clutter motion throughout the coherent processing interval (CPI). These errors thus were suppressed in achieving a DPCA-like filter. Post-Doppler adaptive methods, on the other hand, such as PRI staggered, rely more on Doppler filtering to suppress spectral spreading rather than on adaptivity.

Conclusions 3 and 4 suggested that post-Doppler ADPCA methods apply adaptivity to suppressing clutter confined to a section of the clutter ridge corresponding to the equivalent angular extent of the Doppler filter. Pre-Doppler ADPCA methods apply adaptivity to sharpening the clutter ridge in preparation to a DPCA-like filtering.

An interesting observation was that element space pre-Doppler ADPCA yielded strong apparent target responses at target absent range cells when the target range cell was included in the reference data. The result, with "sliding window" reference data selection, was an apparent loss in range resolution. The target appeared to extend over a number of range cells equal to the window size. This phenomenon might disappear with an increase in number of reference cells when selection methods superior to "sliding window" are implemented.

The three examples presented at the 21 May 1996 TIM demonstrated enhanced filtering resulting from filter method/parameter selection based upon knowledge-based filter environment processing. The MCARM monostatic data (a single acquisition, Figure 3) was used in all cases. Targets, clutter discretes, and jammers were injected into the data as required.

Example 1 (Figure 4) demonstrated that knowledge of potential sources of localized strong clutter discretes, as might be obtained from mapping data, can be effectively applied to enhancing STAP. Subarrays in azimuth were formed from the MCARM element channels (actually, each "element" feeds a column array). The elements within each subarray were linearly combined so that a null in the direction of the discretes was synthesized in each subarray pattern. The weights were computed from a (deterministic) pattern synthesis algorithm. Overlapping subarrays were used so that the subarray size could be large enough to permit synthesizing an adequately broad null with only a small loss of main beam gain. The number of subarrays would be large enough to retain a sufficiently large number of spatial DOF in the subsequent STAP processor. The STAP method was Factored STAP. With STAP alone, the target response was completely submerged by that of the clutter discrete. In combination with pattern synthesis, the target response was 10 dB above the clutter plus noise and the response from the discretes was suppressed to below ambient clutter.



Factored STAP

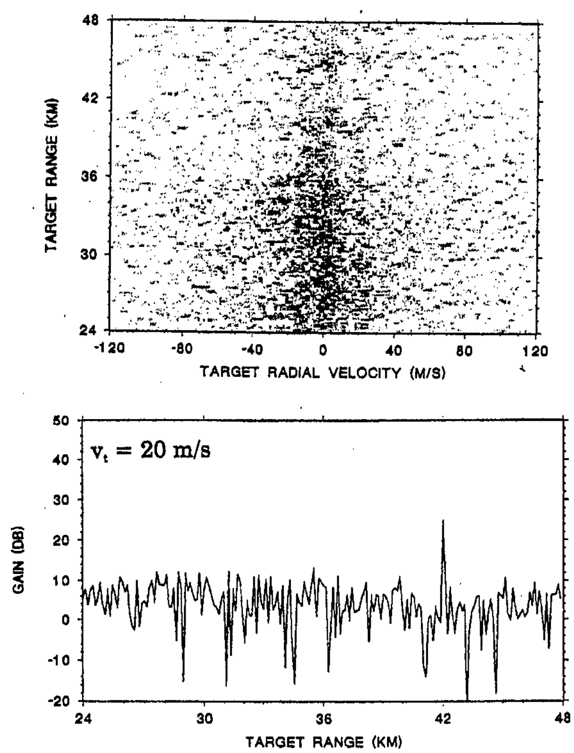
$N_{DOF} = 1 \times 8$

Each spatial channel = 8 "element" subarray

Figure 4: STAP Plus Pattern Synthesis

Example 2 (Figure 5) demonstrated the effectiveness of applying sidelobe cancellation (spatial-only adaptivity) to suppress jamming prior to applying STAP to suppress clutter. The advantage of separating the jamming and clutter suppression adaptive processes is that suitable reference data is far simpler to obtain for jamming cancellation than for clutter cancellation. Jamming cancellation should not be hampered by the difficulties in obtaining adequate reference data for clutter cancellation. In this two-step adaptive process, it is necessary to exclude clutter from the adaptive jamming cancellation step. This was done in Example 2 by using a clutter-free Doppler bin to obtain sidelobe cancellation weights with the MCARM data. Joint Domain Localized STAP was selected with three digitally formed spatial beams and three temporal beams (Doppler beams) for a total N_{DOF} =9. Four auxiliary antennas were created and appropriate jamming and target data vectors were simulated and combined with the MCARM data to obtain weights for canceling jamming in each spatial beam, as in a sidelobe cancellation configuration.

[Alternately, four of the MCARM "element" (column subarray) channels could have been used as the auxiliary channels. The corresponding data would have had to undergo Doppler Filtering to remove clutter from the weight generation process as was done for the spatial beam channels]. The resulting signal-to-interference-plus noise ratio (SINR) was as good as or better than that obtained in the absence of jamming and with only STAP. Apparently the sidelobe cancellers suppress some clutter along with the jamming, allowing for greater overall clutter suppression. An interesting case for comparison would be to combine the four auxiliary channels with the three beams in the STAP process and bypass spatial-only adaptivity. Of course, N_{DOF} would double, a substantial increase that might be unacceptably large for the available reference data.



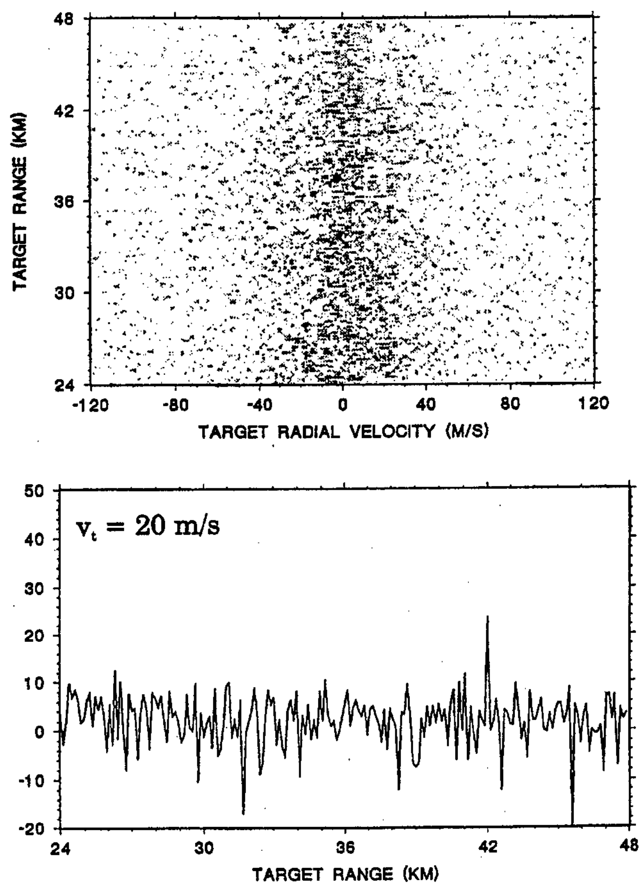
Factored STAP

$N_{DOF} = 22$

N_R = 16 cells (8 from each nonhomo. region)

Figure 5: STAP/Modified Sliding Window

Example 3 demonstrated how mapping data could identify disjoint regions of similar terrain for purposes of increasing the number of reference range cells. Reference data obtained from range cells within these regions could be expected to originate from statistically homogeneous clutter sources. Again, the MCARM data, Flight 5, Acquisition 575, was used. The general terrain corresponding to the data was known to be reasonably homogeneous. Injecting discretes in two regions generated non-homogeneous clutter. Both regions were localized about an angle selected to most interfere with the target. The Doppler from motionless clutter at that angle corresponded to the Doppler generated by the injected target located in the mainlobe direction. The two regions were separated in range, one region including the same range as that of the target. The discretes were not randomly distributed within these bounds. Instead they were carefully selected to enhance the need for obtaining reference data from both regions. In the region containing the range cell of the target, all discretes except one were located along the same angle θ_o; the lone exception was located at an angle θ_o' that differed slightly from θ_o. It was not surprising that use of this region alone for obtaining reference data for STAP resulted in the appearance of at least one false target. In the second region, all discretes except one were located at θ_o' and the lone discrete was located at θ_o. A second false target emerged, not surprisingly, from the application of conventional "sliding window" to the second region. The application of both regions to obtaining training data in searching for targets at ranges within the regions clearly suppressed the two false targets. The STAP technique was Factored STAP. See Figure 6.



128 Pulse Doppler Filter

$N_{DOF} = 3 \times 3$ $N_R = 10$ cells $P_n = 1$ dB $N_a = 4$

Figure 6: JDL STAP/Sidelobe Cancellation (Three 30 dB Jammers)

4.3.2.4 Bistatic Radar Physical Model

A bistatic radar physical model developed by SRC was tailored for use in KBSTAP simulations. Consequently, all required radar environmental modeling software and filtering algorithms became available for developing and assessing knowledge base control architectures. Numerous space-time adaptive and non-adaptive processing (and combinations thereof) of monostatic and bistatic data could now be demonstrated. Data could be imported or locally generated using the monostatic and bistatic physical model software. Imported data could include that derived from measurements or from other physical model simulations, including the AFRL STAP tool.

4.3.2.5 Pre-Adaptive Filtering and Small DOF STAP

At a 19 July 1996 review meeting, it was decided that as much filtering on the data as possible would be done before adaptivity. STAP would then be limited to small DOF approaches, such as Beamspace Post-Doppler Adaptive DPCA. This would allow the available adaptive DOFs to be used where they are most needed. The low DOF approach also has practical advantages of reduced hardware requirements.

An additional disadvantage of large DOF approaches is that angle/Doppler contours vary with elevation angle, and therefore with range. Training data obtained from range cells distant from the target cell would yield weights that are ineffective for suppressing clutter in the target cell. This problem is independent of the homogeneity of the clutter. The problem would be especially severe in the bistatic case; the larger the bistatic angle, the more severe it would be.

In the low DOF approach, knowledge base control focuses on two areas. First, it optimizes the pre-adaptive filtering in space and time, such as with deterministic spatial or temporal pattern synthesis. Second, it optimizes secondary data selection for STAP. Knowledge base control, for example, could ensure that shadowed range cells are excluded from the training data.

4.3.2.6 KBC Demonstration

To demonstrate the above findings within a regime of knowledge base control, a system level example was prepared. A specific architecture regarding the filtering functions was developed. One STAP method, Beamspace Post-Doppler Adaptive Displaced Phase Center Antenna, was selected for the demonstration. This method is a small DOF approach that has been shown to work well by application to MCARM data. Some of the rationale behind this selection was reviewed in a brief technical letter to AFRL dated 02 August 1996. A second technical letter dated 08 August 1996, that described an earlier analysis to confirm and explain filtered interference curves generated by AFRL with MCARM data was also provided.

Only two spatial channels and two temporal channels are employed in the Beamspace Post-Doppler ADPCA method. The spatial channels pertain to the sum and difference beams and the temporal channels correspond to non-delayed pulse stream and one-PRI delayed pulse stream. Two Doppler filter banks are applied to each spatial channel. One filter operates on the non-delayed pulse stream, and the other operates on the PRI-delayed pulse stream. The number of DOF is $N_{DOF} = 2 \times 2 = 4$.

Knowledge base control impacts pre-STAP filtering and selection of STAP reference data range cells. The filtering was expected to have a significant impact in performance because of the small N_{DOF}. Both Doppler filter weighting and spatial filter (beam) weighting included sidelobe pattern synthesis to selectively suppress potentially strong interference in known locations in angle and Doppler. These locations, for example, could correspond to the intersection of a highway and a range cell of interest. The orientation of the highway with respect to the radar platform velocity vector and the probable speeds of vehicle traffic, especially large cross section trucks, would be useful parameter data for specifying the synthesis procedures. In general, the synthesis weight computation and STAP weight computation would be repeated for each target Doppler and range.

Also, the spatial synthesis would be carried out simultaneously with spatial-only adaptive nulling in the event that sidelobe jamming is sensed. The synthesis procedure discussed in the DP [3] was well suited for simultaneous adaptive and deterministic nulling.

The antenna configuration proposed for the example was a large S-band rectangular array with 16 rows and 216 columns of elements. Analog combiners would be used to divide the array into nine sum-beam and nine difference-beam manifolds. The array would be fully populated with phase shifters to retain low sidelobes with scan over a wide field-of-view in azimuth and limited field-of-view in elevation. Three manifolds would provide uniform illumination for transmit, 35 dB Taylor azimuth and 35 dB Taylor elevation for the sum-beam, and 35 dB Bayliss azimuth and 35 dB Taylor elevation for the difference-beam. A receiver and A/D converter would be associated with each sub-manifold. The nine digital channels for each receive beam would provide the flexibility required for the deterministic and adaptive sidelobe nulling procedure. Subsequent digital processing would include deterministic notching to suppress clutter discretes in both the space and time (Doppler) domains, spatial-only adaptive nulling to suppress sidelobe jamming, and post-Doppler beam space (sum and difference beams) ADPCA STAP to suppress homogeneous clutter. The antenna/processor configuration would have the advantage of large AWACS-like size (3,456 elements) with only a moderate number of receivers (18).

RLSTAP would be used to generate "data cubes" corresponding to the 18 digital channels. Detailed clutter simulation that includes the effects of varied terrain scattering and shadowing would be possible. An important consideration was the facility with which RLSTAP could accommodate the receive antenna subarray parameters corresponding to a data cube. Each subarray would be a 16 x 24 = 384 element array with 35 dB Taylor weighting applied to the columns, and a part of either a 35 dB Taylor or a 35 dB Bayliss weighting applied to the rows. Each element would also contain a phase shift corresponding to the desired beam direction, and each subarray would be centered at one of nine different locations with respect to the reference origin of the entire antenna. The complex weighting of the subarrays or the corresponding complex subarray patterns could be generated. If feasible, it was preferable that the weighting option could be exercised with RLSTAP, so that the precise pattern values in the clutter cell directions could be used in computing clutter signals. An alternative method, inputting patterns to RLSTAP, would require a two-dimensional complex interpolation between pattern points.

At a 15 October 1996 TIM, SRC presented the results of simulations based on this 18-channel, large planar array antenna and filtering processor depicted in Figure 7. The filtering example that was presented demonstrated the potential of deterministic nulling. Post-Doppler beam space ADPCA STAP was shown to be a potential high-payoff STAP method with these advantages:

- ADPCA enables nulls to best align with clutter ridge
- Small numbers of DOF implies manageable number of receivers
- Only $\Sigma$and Δ beams required; no "extra" beam forming required
- Small number of DOF implies small number of reference cells
- Pre-filtering (beam space/post-Doppler) allows most effective application of STAP DOF
- Note: knowledge based controller optimizes pre-filtering

The demonstration model was subsequently enhanced further to include a large degree-of-freedom STAP option to augment the low degree-of-freedom option. The large DOF option uses a configuration of beams and subarrays in a two-pulse adaptive displaced phase center antenna algorithm. The large DOF method would be selected when notching is not required.

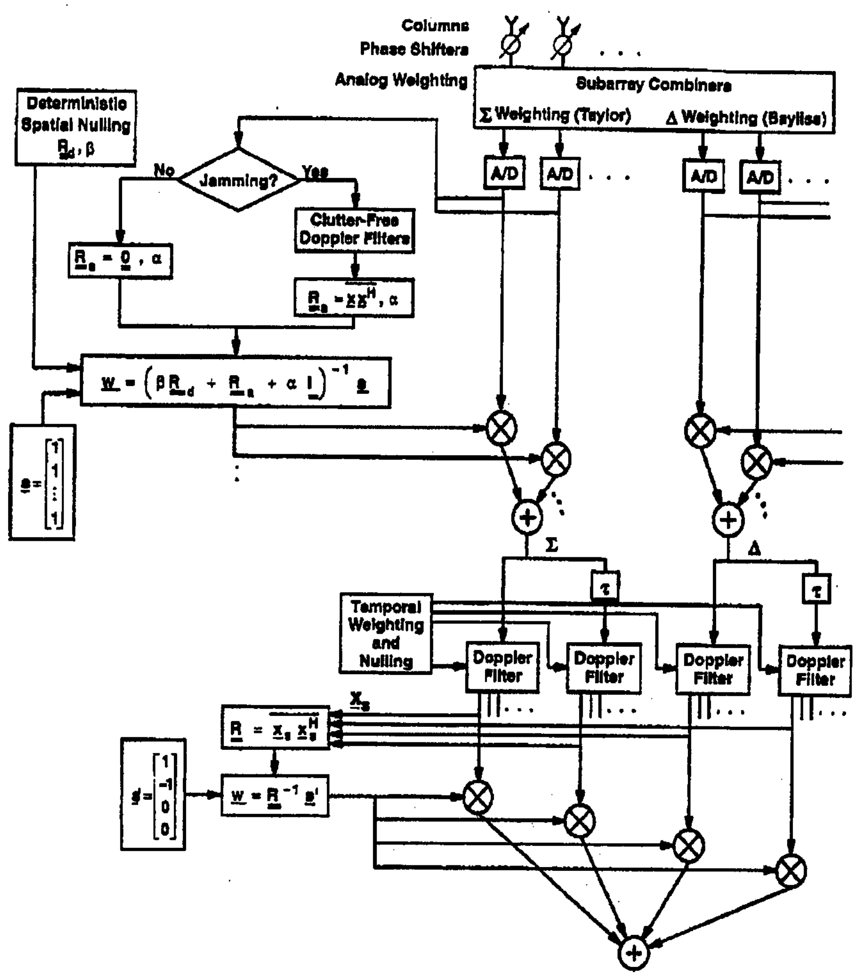


Figure 7: Antenna/Filter Architecture

4.3.2.7 Sample Selection

Three methods of sample selection were implemented and applied to MCARM data (Flight 5, Acquisition 575) for demonstrating comparative effectiveness: Generalized Inner Product (GIP), Sample Matrix Inversion (SMI), and Generalized Likelihood Ratio Test (GLRT). GIP and SMI were implemented as two-pass techniques. The GIP and (CFAR normalized) SMI test statistics were generated from a sliding window sampling of data throughout a range extent of interest. Non-homogeneities were identified by thresholds in SMI and rank ordering in GIP. Finally, the data in both cases were filtered by CFAR normalized SMI weights obtained from all samples in the range extent excluding "non-homogeneities," test, and guard cells. GLRT on the other hand was applied in a single pass structure (the maximum likelihood ratio - under the assumption of Gaussian interference - was computed from a sliding window sampling) because it was found that a two-pass application of GLRT did not yield significant improvement over the single pass. It was found that GIP out-performed SMI, but that GLRT (without an explicit non-homogeneity detection pass) was superior to both. GLRT appears to be naturally insensitive to non-homogeneities, although this required further investigation. Only a very limited data set was used in this research.

Sample selection for adaptive algorithms was identified as high payoff area in a 06 May 1997 TIM. The evaluation example shown in Table 5 was used to examine the following methods:

- SMI sliding window
- SMI two-pass (non-homogeneity detection)
- GIP/SMI (non-homogeneity detection)
- GLRT sliding window
- GLRT two-pass (non-homogeneity detection)

Data: MCARM Acquisition 575/Flight 5
Injected Target
››Range Cell 290
››Range rate 18.5 m/s (in mainlobe clutter) 128 Pulse Hanning Weighted Doppler Filter
Factored STAP
››$N_{DOF} = 22$
››±2 Guard cells

Table 5: Sample Selection Evaluation Example

Table 6 lists the recommended selection methods or rules yielded by these experiments

Method	Specifications
GLRT	▪ $K \approx 4\ N_{DOF}$ ▪ No diagonal loading
GIP/SMI	▪ **Preferred method** ▪ Adjust lower/upper GIP threshold such that $K \approx 3\ N_{DOF}$ ▪ No diagonal loading ▪ Fixed target detection pass
SMI Two-pass	▪ Adjust threshold in NHD pass to accept ~ 80% of samples ▪ ~ 25% diagonal loading ▪ Sliding window target detection pass; $W \approx 1.5\ N_{DOF}$

Table 6: Rules for Sample Selection Methods

Possible benefits to sample selection that could be derived from knowledge of shadowed terrain were also demonstrated at the TIM. Table 7 defines 12 examples that were performed using GIP/SMI in known regions of shadowing and discretes.

Sample Selection - Shadowing Example ▪ **Data: MCARM Acquisition 575/Flight 5** ▪ **Target at Cell 290** ▪ **GIP/SMI with fixed window target detection pass (L = 150; 245 – 395) and no diagonal Loading**	
Shadowed cells distant from test cell and nearby discretes ▪ Shadowed Cells 350 –395 ▪ Discretes ~ 300	
▪ Lower/upper thresholds 2.8 N_{DOF}/6 N_{DOF}	1. NHD pass included shadowed cells, average K in target detection pass = 47 2. NHD pass excluded shadowed cells (unsymmetrical window), average K in target detection pass = 44
▪ Lower/upper thresholds 2 N_{DOF}/6 N_{DOF}	3. NHD pass included shadowed cells, average K in target detection pass = 78 4. NHD pass excluded shadowed cells (unsymmetrical window), average K in target detection pass = 75
Shadowed cells adjacent to nearby discretes ▪ Shadowed Cells 300 –350 ▪ Discretes ~ 300	
▪ Lower/upper thresholds 2.8 N_{DOF}/6 N_{DOF}	5. NHD pass included shadowed cells, average K in target detection pass = 48 6. NHD pass excluded shadowed cells (unsymmetrical window), average K in target detection pass = 49
▪ Lower/upper thresholds 2 N_{DOF}/6 N_{DOF}	7. NHD pass included shadowed cells, average K in target detection pass = 66 8. NHD pass excluded shadowed cells (unsymmetrical window), average K in target detection pass = 67
Shadowed cells adjacent to test cells ▪ Shadowed Cells 291 –340	
▪ Lower/upper thresholds 2.8 N_{DOF}/6 N_{DOF}	9. NHD pass included shadowed cells, average K in target detection pass = 52 10. NHD pass excluded shadowed cells (unsymmetrical window), average K in target detection pass = 50
▪ Lower/upper thresholds 2 N_{DOF}/6 N_{DOF}	11. NHD pass included shadowed cells, average K in target detection pass = 72 12. NHD pass excluded shadowed cells (unsymmetrical window), average K in target detection pass = 71

Table 7: Examples of Sample Selection in Shadowed Terrain

These experiments suggested the following rules for sample selection in regions of shadowing:

- Adjust GIP lower threshold (to compensate for reduction in number of samples due to shadowing) such that the number of samples for filtering (K) $\approx 3\ N_{DOF}$

- Shift GIP sliding window (in NHD pass) to avoid shadowed region (yield some benefit if distributed strong discretes reside near shadowed region)

4.3.2.8 Spatial Notching

Notching was identified as a high payoff area in the 06 May 1997 TIM. Spatial notching was noted as being especially suited for mitigating false alarms caused by moving interference that fall within the Doppler cell under test, and combining well with pre-STAP sidelobe jamming cancellation. Architectures and examples of Small N_{DOF} STAP and Large N_{DOF} STAP were demonstrated at the TIM. The following process and bases for rules for performing effective spatial notching were suggested:

- Determine air and ground traffic interference that lie in the range/Doppler cell under test

- Prioritize interference

- Determine notch widths

- Sense jamming

- Apply Small N_{DOF} STAP with notching and/or jamming cancellation with heavy Doppler weighting

Highway traffic as a source of interference was discussed in detail. It is a potential source of interference if a highway crosses a test range cell and the radial velocity (range rate) lies within the test Doppler cell, i.e., when:

$$[V_t - W_d/2,\ V_t + W_d/2] \cap [V_o \sin(\theta_d + \theta_c) - V_d\,|\sin\alpha|,\ V_o \sin(\theta_d + \theta_c) + V_d\,|\sin\alpha|\,]$$

is not empty, and:

V_t = Target radial velocity (including ground component)

V_o = Platform speed

θ_d = Angle to highway crossing

θ_c = Crab angle

α = Angle subtended by highway and range cell

W_d = Doppler filter width [$W_d = \beta_d \lambda / 2T_{CPI}$ where β_d = Doppler beam broadening factor (function of sidelobe weighting)]

Figures 8 and 9 illustrate the geometry of highway interference and notching placement. A notching priority parameter was also introduced as:

$$\eta = R^{-4}(\theta, \phi)G_t(\theta, \phi, \theta_o, \phi_o)Gr(\theta, \phi, \theta_o, \phi_o)f(\alpha)$$

where:

$G \quad = g_x g_y$

is the gain pattern (column, row weighting) and:

$g_x \quad$ = Horizontal plane gain pattern

$g_y \quad$ = Vertical plane gain pattern

In contrast with scattering from distributed clutter sources, it was pointed out that scattering from discretes:

- Does not exhibit well defined grazing angle dependence
- Exhibits R^{-4} (for monostatic) range dependence (distributed clutter is closer to R^{-2})
- May be a well defined function (f) of angle (α) between the highway and range cell

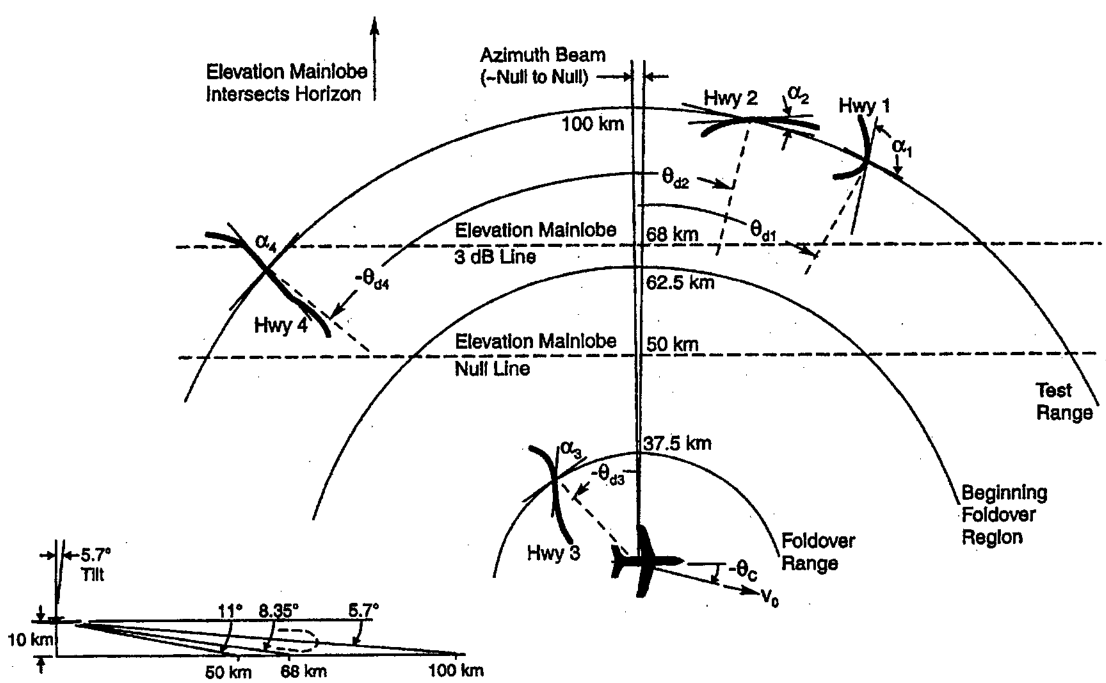


Figure 8: Notching Geometry

It was noted that the notch width for highway traffic is dependent on the angle between the highway and range cell and maximum notch depth (limited by antenna errors and sidelobe noise floor). The notch width may also be chosen to encompass highway crossings in near range cells; these cells can then function better as reference cells. A rough relation for the number of required spatial channels is :

$$N_s = 10^{A/20}(\Delta\pi D\cos\theta_d)/2\lambda$$

where:

A	= Notch depth (dB below quiescent sidelobe level)
θ_d	= Angular location of notch from broadside
D	= Aperture extent in horizontal plane
Δ	= Notch width (radians)
λ	= Wavelength

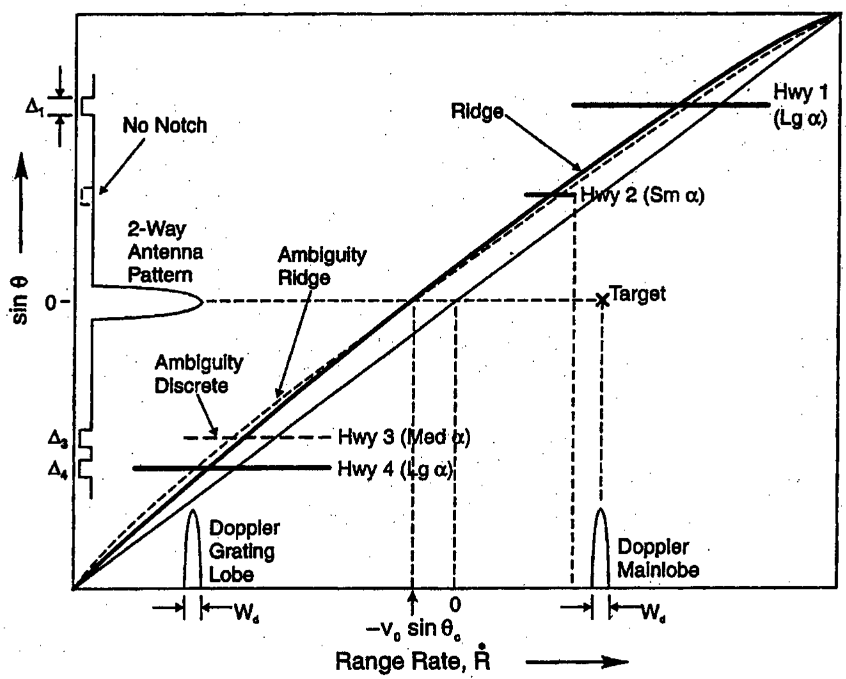


Figure 9: Notching Placement

A review of STAP algorithm evaluation resulted in a fine adjustment of the STAP algorithm selection rules. A principal find was that combining beams and subarrays into a set of spatial channels results in a superior STAP algorithm architecture. This set could be implemented in both Factored STAP and ADPCA configurations. The former is more effective for mainlobe clutter suppression with a large number of channels. The latter is preferred when the number of channels is limited. A set of relations was derived that would be used by the knowledge-based controller to determine numerical values in selecting STAP algorithms. These values quantify the boundary between mainlobe and sidelobe clutter and the proximity of the Doppler lobe (or ambiguity) to mainlobe clutter.

4.3.2.9 Alternate Radar Architecture

An alternate architecture and rules for its optimization were incorporated into KBSTAP. The architecture is characterized by at least one receiver per column of array elements and permits digital subarraying in space and in time (forming subarrays and "sub-CPI's") that is selected by the KBC to optimize interference filtering. The filter can be adjusted for each test Doppler cell. In comparison the original architecture assumed a hardware "manifold" subarraying and did not offer the same flexibility.

Two modes were identified. In both modes the subarrays and sub-CPIs are determined non-adaptively (with the exception of spatial-only adaptivity to cancel jamming). In Mode 1, the array factor and "CPI factor" are determined by STAP methods following the rules for the original architecture. In Mode 2, the array factor and CPI factor are determined non-adaptively, again using rules as per the original architecture.

In the new architecture, the subarrays/sub-CPIs are determined jointly to null most of the clutter ridge (Modes 1.2 or 2.2) or independently to null only mainlobe clutter in space and time (Modes 1.1 or 2.1). A set of rules in terms of an iterative procedure was determined for optimizing the subarray/sub-CPI sizes and array-factor/CPI-factor sidelobe weightings for Mode 2.1. These parameters are determined such that two performance constraints are met. One constraint is a level of suppression of clutter below noise. It often is desirable to suppress clutter well below noise (by 10 or 15 dB) because of the large uncertainty in clutter statistics. The greater the clutter suppression, the easier it is for the detection processor to meet specified detection and false alarm probabilities. The second performance constraint is the "processing margin," the allowable signal to interference plus noise degradation resulting from interference suppression procession. If the margin cannot be met, the filter processor reports a degraded probability of detection to the KBC. The rules for digital subarraying aim at optimizing interference rejection within constraints on signal-to-noise margin and angle and velocity. An architecture shell was added, with stubs for filtering, detection, and tracking functions. Work on the individual functional elements was accomplished separately.

4.3.2.10 Filtering Rulebook

SRC completed a Radar Filtering Rulebook (Volume III) in October 1997, which brought together and formalized six (6) knowledge based rules that had evolved through the course of the filtering research discussed above. The description for each rule is comprehensive and includes:

- A technical discussion
- A list of data sources required for decision making ("KBC input")
- A list of the filter parameters that must be specified by the KBC ("KBC output")
- The decision process and mathematical development ("KBC algorithms")

Table 8 encapsulates the six (6) published rules

Rule	Method	Description
1	Subbarrays/Sub-CPI Nulling	Non-adaptive method whereby optimized space/time subarrays are formed that "pre-filter" interference prior to adaptive filtering (e.g., STAP)
2	Space/Time Sidelobe Levels	Determining the optimum receive array weighting and Doppler filter weighting to apply in deterministic filtering and in the steering vector for STAP filtering
3	STAP Selection	Selection of the optimum STAP method to apply, or whether to apply STAP at all
4	Array Notching Plus Spatial Only Adaptivity (Antijam)	Deterministic notching of the receive antenna array factor for purpose of suppressing moving discretes type clutter (aircraft, highway traffic, etc.) from the test and reference range cells applied in STAP
5	Sample Selection/Shadowing	Optimum means of choosing reference range cells for STAP
6	Diagonal Loading	Determination of the minimum amount of diagonal loading required for STAP

Table 8: KBC Filter Rules

Specific filter processors are described in an associated SRC report entitled "Airborne Radar Filtering." This report was delivered as three standalone sections, references [6], [7], and [8]. Examples contained in those references were used in part in determining the rules described in the rulebook. The first section subtitled "Space/Time Adaptive Processing" [6] is a summary of STAP features that became apparent during the KBSTAP effort to identify knowledge base control rules for radar filtering. It groups practically all variation of STAP into eight general methods given here as:

1. Factored
2. Element Space Post Doppler ADPCA
3. Element Space Pre Doppler ADPCA
4. Beam Space Post Doppler ADPCA
5. Beam Space Pre Doppler ADPCA
6. Joint Domain Localized
7. SLC Post Doppler ADPCA, (Σ Δ, & Subarrays Post Doppler)
8. SLC Pre Doppler ADPCA, (Σ Δ, & Subarrays Pre Doppler)

The section discusses the processing characteristics of these methods and illustrates each of them with a block diagram. The STAP methods were applied to the MCARM monostatic radar data made available through AFRL. The goal was to identify rules regarding the relative performance of the methods in suppression of airborne radar clutter. Table 9 defines the MCARM data and radar environment used by SRC to assess the STAP methods.

Example Parameters	
Attenna	Side mounted area with 24 combiners on receive
Channels	22 subarray, a Tayler Weighted Σ , and a Bayliss Weighted Δ
Subarray channels	Two horizontal rows of 11 subarrays.
Subarray	Four radiating elements aligned in elevation. Rows are displaced in elevation (one directly above the other) by the height of the subarray
Frequency Band	L-Band
Range Resolution	120 m.
CPI	128 PRIs
Aircraft Velocity	100 m/s.
Altitude	10,000 ft.
Crab Angle	7.28°
Terrain	Vicinity of the expanse between the Chesapeake bay and the Delaware River at about the latitude of Baltimore
Injected Target Signal	At range cell 350 Radial velocity 20 m/s

Table 9: MCARM Data and Radar Environment

Results and conclusions stated in this document are bulleted below. These conclusions are discussed and supported more fully in the reference [6]:

- Post Doppler beam space ADPCA methods generally were found to be more effective than pre-Doppler or element space ADPCA methods especially for small N_{DOF}.

- In general, it has been found that Doppler filtering and beam forming, all "deterministic processing," best precede STAP adaptivity

- When STAP precedes pulse compression pulse compression, STAP performance generally is best if STAP also precedes Doppler filtering.

- Whereas Factored STAP losses effectiveness if the Doppler filter sidelobes levels rise above the two way antenna gain pattern sidelobe levels, element space ADPCA is actually more effective with the higher Doppler sidelobe levels

The second standalone section [7] of the SRC report discusses subarray/sub-CPI nulling and recommends a revision of Rule 1 as published in the Rulebook [5]. In this section, subarray/sub-CPI filtering is discussed along with an iterative method for determining the optimum sizes of the subarrays and sub-CPIs. This method forms the basis of two knowledge base controller rules for deterministic filtering (Rules 1 and 2 in Reference [5] and Table 9). The aim of the method is to suppress mainlobe clutter (in space and in time) to the array factor and CPI factor sidelobe levels within the constraint of maintaining a minimum signal to noise ratio.

A variation of Rule 1 was investigated that may prove superior to that reported earlier in Reference [5] (Table 9). In this variation, the subarrays and sub-CPIs are sized to maximize an estimate of signal to clutter ratio within the constraint of maintaining a minimum signal to noise ratio. This attractive variant is discussed in the concluding example subsection. The referenced report section [7] is well illustrated with figures showing the example and results of the procedure.

The third standalone section [8] of the SRC report provides the development of array notching and spatial-only adaptivity (Rule 4 of the rulebook [5] and Table 9). Pattern synthesis (deterministic notching) and spatial only adaptivity in combination with a small degree of freedom STAP was demonstrated to be advantageous in suppressing most types of interference. An alternative filtering architecture that allocates all available spatial DOF to STAP appeared to work very well also, perhaps even better, except for the potential of false alarms arising from the discretes. The tracking process could discard these false alarms so they are probably not major concern. The major benefit of deterministic notching appears to be that it is applied with a small DOF STAP method with the associated requirement for less reference data than are required for the large DOF STAP methods. Notching and "spatial only adaptivity" are fully discussed in the reference [8] and an example used to validate and demonstrate the approach is described. As in the other separate sections of the SRC report the document provides much greater detail than is provided here.

4.3.2.11 Filtering Processor Results

We studied a few simulation scenarios to evaluate the knowledge base approach to control system parameters. The scenarios were simulated by injecting targets in MCARM flight #5 data. The experiments included the influence of close-by discretes on the detection of the main target. In the first experiment, we studied the influence of having a discrete or another target in the close vicinity of the main target. In order to find out the advantage of the knowledge-base approach, we wanted the separation between the main target and the discrete (or the secondary target) to be less than the length of the sliding window on either side of the cell under test. This is a case where the training cells of the covariance matrix contain a discrete or another secondary target. In that case, the training cells are contaminated and do not represent the statistical behavior of the background terrain clutter. This is a case where the conventional symmetrical sliding window would not give the best results. The assumption —in the case of contaminated training cells — is that the non-homogeneity detector should be able to detect the out-lier and exclude it from the training cells constructing the covariance matrix.

A simulation program was written to show a hypothetical scenario of a primary target and a secondary target flying with similar speeds in the same direction. This assumption puts the primary and secondary targets in the same Doppler bin, but at different range bins. Notice that the secondary target does not have to be a real target, it could be a strong discrete in the side-lobe clutter, or it could be a reflection coming from road traffic with the same radial speed as that of the primary target. To study the difference in performance between the conventional and knowledge base approaches, we selected the Factored STAP as the testing algorithm. In the conventional case, we took the training cells from a symmetrical window around the cell under test. In the knowledge-based approach, we ran the non-homogeneity detector on the secondary data before selecting the training set. The non-homogeneity detector recognized the discrete (secondary target) as an out-lier and excluded it from the training set. For the purpose of the experiment, we varied the input SINR of the primary target and measured the output SINR. Figure 10 shows that the knowledge-based approach with the non-homogeneity detector outperforms the conventional approach with the symmetrical sliding window with an average of 5 dB gain. The test used MCARM flight #5 data.

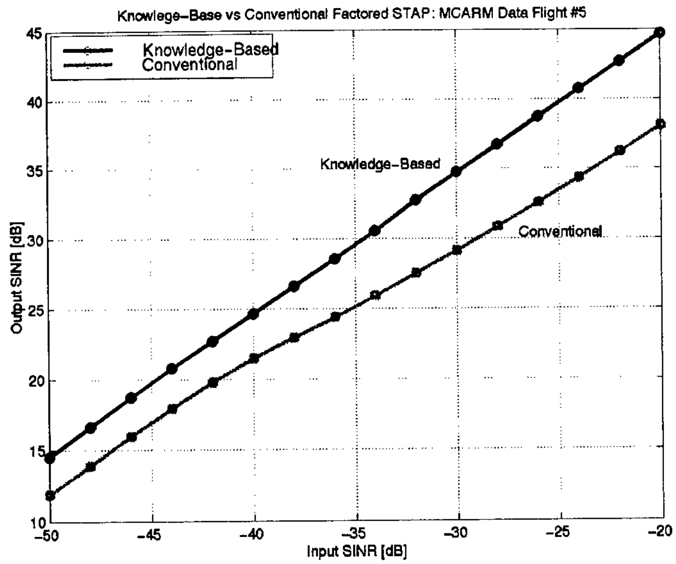


Figure 10: Knowledge-base And Conventional Factored STAP

The performed experiment indicates that the knowledge-based approach was not affected by the presence of a discrete (or secondary target) in the near vicinity of the cell under test. In addition, the presence of a discrete (or secondary target) in the vicinity of the cell under test causes a degradation in the output target level using the conventional Factored STAP algorithm.

Another experiment was performed to discover more effects caused by a secondary target or a discrete in the vicinity of the primary target. This time, we fixed the power of the primary target and changed the power of the secondary target. Figure 11 shows the comparison between knowledge-based and conventional Factored STAP approaches.

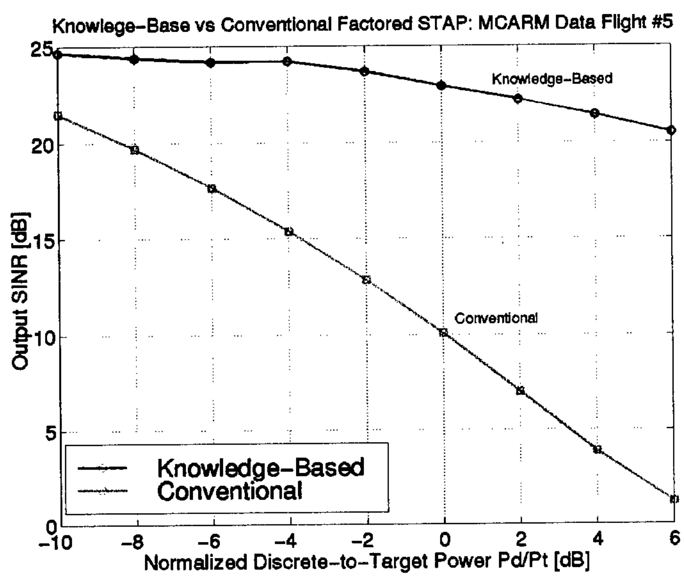


Figure 11: Knowledge-base versus Conventional Factored STAP Approaches

The horizontal axis in Figure 11 is the power ratio between the secondary target and the primary in [dB]. The impact on the primary target is not severe when the power of the secondary target is small relative to the power of the primary target. However, when the power of the secondary target starts increasing compared to that of the primary target, the influence becomes more severe. At some point, the primary target becomes completely undetectable.

4.3.3 Knowledge Based Tracker

The following subsections cover the actions taken and accomplishments toward the development of a knowledge-based tracking proof of concept demonstration and a tracking subsystem that could be integrated into an end-to-end KBSTAP system. The primary question to be answered was the degree to which knowledge about the local radar environment could enhance the ability of the tracking processor to maintain multiple tracks through areas of shadowing, clutter, and other sources of interference that might be located. As with the filtering process, a goal was also to identify working, knowledge-based rules that would guide the tracker based on available sources of information. TSC was primarily responsible for the research and development on the Knowledge Based Tracker (KBT) component of KBSTAP.

4.3.3.1 Scenario Generator

Work on the knowledge-based tracking process began with the identification and description of key support elements. These included a multiple target-track scenario generator, a primary radar report generator, a non-collocated IFF report generator, a functioning tracker with data association and maneuver detection logic, and a figure of merit capable of describing the overall tracking performance. In addition, a set of initial computer experiments was proposed to test and verify existing control rules and provide insight into additional improvements.

The scenario generator was capable of simulating maneuvering targets whose trajectories were composed of linear segments and centripetal arcs. The White Sands Missile Range (WSMR) was used as a location for developing a scenario for an end-to-end example requested by AFRL at the 21 May 1996 TIM. WSMR was selected because of the immediate availability of Defense Mapping Agency (DMA) data, and because measured data is available from the RSTER array radar. An RSS map of WSMR allowed the definition of a reasonable scenario for a high priority, low observable target. The selected scenario would fly through regions of changing elevation, clutter, obscuration, and multi-path conditions. To deal with complexity, an additional maneuver trajectory feature was added to the scenario generator, which allows complex trajectories to be built up from linear, non-maneuvering segments, plus accelerating segments (centripetal only), consisting of turns about a designated center of curvature.

A prototype WSMR trajectory was designed, which contains several features pertinent to the knowledge base tracking concept. The trajectory includes both linear and evasive type maneuver track sections in order to test and adapt tracking filter response. Maneuvers were modeled as 2g centripetal accelerations with 40 degree turn angles. A linear portion of the track was situated perpendicular to the radar line of sight to create a zero Doppler region. A capability of including clutter and multi-path corruption of the tracking data was also included.

The capabilities of the scenario generator are itemized in Table 10. Figure 12 illustrates the scenario generator's graphic user tools for generating target tracks.

	Scenario Generator
Purpose	• Allows user to specify target tracks in x, y position and velocity, and simulate radar measurements • All trajectories composed of altering ▪ Linear non-maneuvering sections ▪ Circular arc (centripetal) maneuvering services
Inputs	• Radar scan time • SNR at mid-scenario range • Target speed • End points of each section (mouse driven) • Centripetal acceleration and turn angle of each maneuvering section
Outputs	• True track positions and velocities x, y, v_x, v_y sampled at radar scan rate • Corresponding measurement noise corrupted track report information including: ▪ x, y, v_x, v_y ▪ SNR ▪ Range/cross range uncertainties ▪ State covariance matrix ▪ Detection status (assumes SWI target statistics)

Table 10: Scenario Generator Capabilities

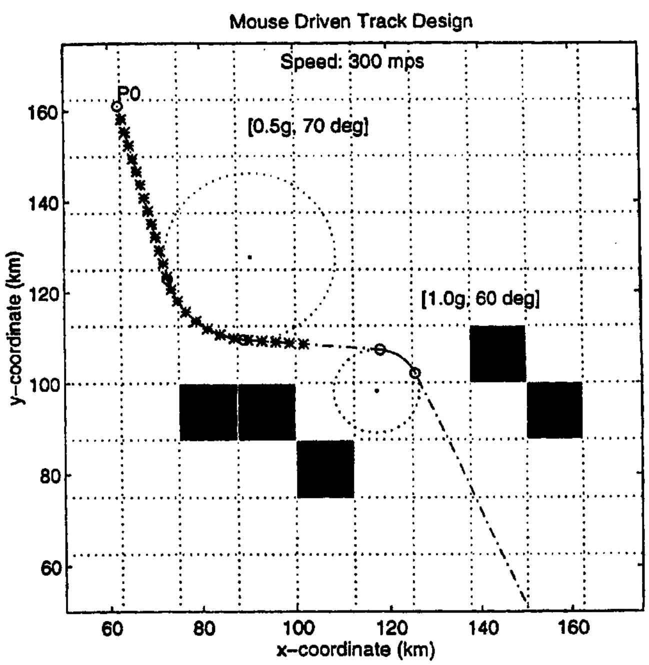


Figure 12: Scenario Generator Track Design

4.3.3.2 Track Prediction Error

The effects of maneuver anticipation on track prediction error were examined. Two types of detection gates were applied. In the absence of any detected maneuver, the uncertainty gate was assumed to be an ellipse oriented along the range and cross range directions. Each semi-axis of the ellipse was composed of three times the root-sum-square of two components: the radar measurement uncertainties, already in range and cross-range coordinates, and the predicted Kalman x and y position uncertainties, projected into the range/cross-range coordinates. When maneuvers are detected or anticipated, a constant speed centripetal acceleration maneuver gate is used. Nominally, this gate assumed accelerations of between 0.5 g and 2.0 g. In a coordinate system with the instantaneous track direction oriented along the x-axis, the maneuver envelope forms a chevron shaped structure. For this case, detection was assumed to occur whenever the aforementioned uncertainty ellipse intersects or falls within the maneuver chevron.

4.3.3.3 Tracking Rule Development

Knowledge base rules were developed to adapt the Kalman tracker to target maneuvers. The metrics that were used to flag maneuvers depended on the differences between measured and estimated position and acceleration values. Specifically, the maximum of the absolute x and y component deviations for both position and acceleration were compared to thresholds after each scan. If both position and acceleration thresholds were exceeded, the maneuver covariance matrix was increased in a manner consistent with a 2g maneuver having a decorrelation time that could vary from between one hour and one second. Initial results showed that maneuver induced track position errors could be reduced from several kilometers down to a few hundred meters.

A maneuver anticipation rule was developed and tested. Using the scenario generator, a track was constructed consisting of a linear approach, a 1 g centripetal turn, a second linear segment, a 0.5 g centripetal turn, and a final linear segment. Using a priori "map" information of the target's proximity to an obstacle, 1 g of maneuver noise was added into the tracker's plant covariance matrix several scans before the tracker would normally respond to the target's maneuver. The resulting tracking performance of this proactive tracker was seen to be superior to the normal reactive tracker operating on the same data. Not only were the peak, mean, and standard deviation of the tracking error smaller, but the stability of the proactive tracker was also better.

Rule development also explored issues of priority-dependent track-state promotion logic, track bifurcation termination, and association logic for the multi-target tracking algorithm. Association rules based on proximity of reports with track gate center, track quality, and track SNR history were considered. These rules were and published in a KBT Rule Book (Volume V). See the following sections.

4.3.3.4. Development of a Working KBT

TSC briefed the to-date progress on the development of a KBT at a 15 October 1996 TIM. At this time TSC presented a KBT design overview summarized in Table 11 and a high level functional flow reproduced as follows:

1. Detection validation: test for discretes, ground traffic, etc.

2. Track initiation: Associate detection reports until confirmed track state is achieved

3. Predict position and velocity: Using Kalman or $\alpha\beta$ tracker, supply target information to database

4. Potential maneuver contour: Determine locus-of-points that defines physically possible target maneuvers

5. Anticipated maneuver conditions: Evaluate database to identify conditions for which target maneuvers will probably occur. Adapt KBT by increasing gain or bifurcation track prior to maneuver

6. Deterministic maneuver conditions: Evaluate database to identify conditions for which target maneuvers must occur. Position data-to-track association windows accordingly

7. Reduce detection thresholds: Evaluate limited portion of data with lower threshold if detection of high-priority target does not occur

8. Coast/demote track state: Base decision for track state demotion on database information (CFAR level, Obscuration, etc.)

9. Track-before-detect processing: Maintain TBD processor on all high-priority targets

10. Update database: Performed at end of every radar scan

The KBT Consists of Four Components	
Post-Detection Processor	• Interface between detection processor and KBT • Manipulates Data Into Format Usable by Tracker • Centroid Data Collection from Multiple CPIs • KBT can request Reduced Threshold Detection Reports
Data to Track Association Logic	• Pairs detection reports with existing Tracks
Tracking Filter	• $\alpha\beta(\gamma)$or Kalman Filter Algorithms
Track State Promotion Logic	• Monitors Quality of Existing Tracks
The KBT will Interact with KBSTAP by:	
• Receiving detection reports from the detection processor • Evaluating reports in context with information from the database • Merging data with existing tracks • Reporting through the database and KBC	

Table 11: KBT Design Overview

At this time, TSC also presented examples comparing reactive (conventional) tracking with the proactive (knowledge-based) tracking that uses a maneuver anticipation rule derived from known terrain and probable objectives. Numerous x-y coordinate plots of predicted versus measured locations with ellipses of uncertainty, predicted errors, and peak absolute errors were used to illustrate the relative tracking performance. Figure 13 depicts examples of these illustrations selected for the same segment of track. Observations from this comparison are shown in Table 12.

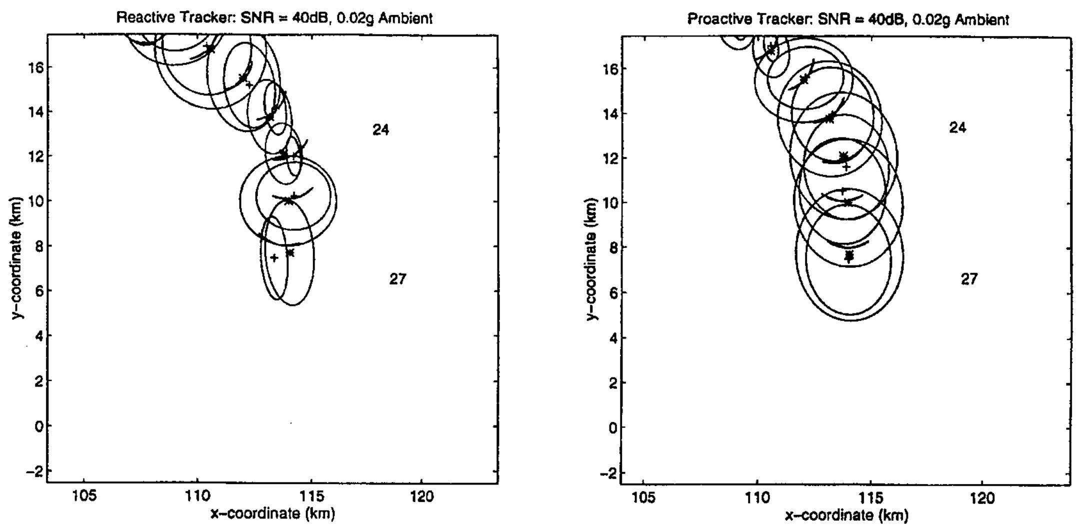


Figure 13: Reactive Versus Proactive Tracking

Rule Implementation Example

Compare Reactive (Conventional) Tracking with Pro-Active (Knowledge Based) Tracking:

Apply Maneuver Anticipation Rule

Target Scenario: Low Flying Aircraft Turns to Avoid Elevated Terrain

Observations:

Reactive Tracking	Pro-Active Tracking
• Uncertainty Ellipses Remain Small Before Maneuver Begins • Position Errors Become Large in Turns • Maneuver Gate Triggered by Poor Prediction of Target Location • Kalman gain Remains High for Significant Length of Time	• Kalman gain Increased on Scan 9 • Ellipse Area Initially larger than for Reactive tracker • Ellipse Area Remains Lower • Quick Recovery After Maneuver (No Track Oscillations Observed)

Table 12: Knowledge Based Tracker Example

At the 06 May 1997 TIM, TSC announced a working (at least as a self contained test-bed) multi-tracking KBT with the following capabilities:

- Process reports on a per scan basis
- Correlates reports with existing tracks
  - Forms binary gate correlation matrix
  - Forms distance matrix
- Updates track status
  - Dropped
  - Tentative
  - Firm
- Updates track quality/age
- Creates new tentative tracks with unused reports

Figure 14 is a high-level flow diagram of the multi-target tracker as presented.

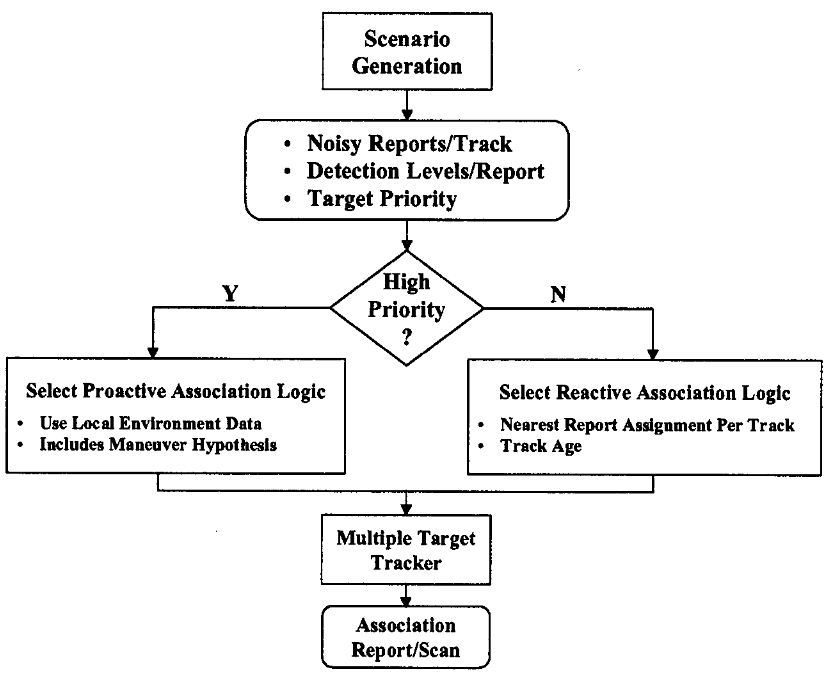
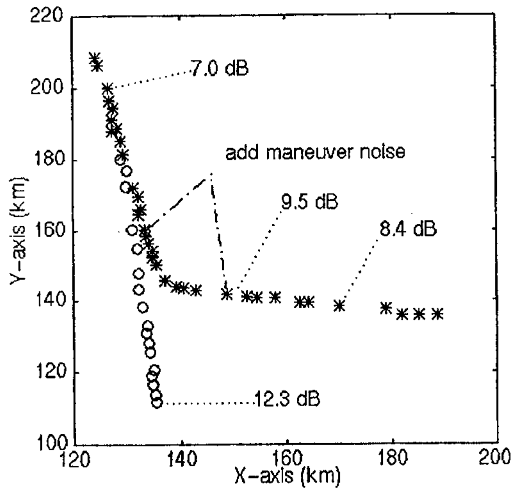


Figure 14: Tracker Flow

Four case studies, which demonstrated the above capabilities, were presented at the TIM. Table 13 defines the four cases and their detailed features. Case 3 included both maneuvers and multiple targets and was tracked using a reactive tracker and two strategies for proactive tracking. Figures 15 and 16 plot the detection input and the resulting track using the proactive strategy #2.

CASE	TYPE	FEATURES
1	Multi target tracker example	• Two crossing targets ▪ Straight-line flight paths ▪ ~ 15 dB SNR ▪ Swerling case 1 fluctuations with drop outs ▪ 200 m/s target velocities • Radar operations ▪ Two reports per scan (max) ▪ No false alarms in this case ▪ 10 second update rate » 2 km target motion per scan ▪ Multiple detection levels » Used to test and validate KB track rules
2	Actual radar data example	• Tested multiple target tracker using recorded radar data ▪ Included false alarms ▪ Position-only data ▪ Measured uncertainties unknown • Example helped to validate tracker performance • Alternate hypothesis formed three times • False track briefly formed and dropped
3	Multi-target tracker example	• Initial primary target launches secondary target then maneuvers ▪ Primary pulls 0.5g/70° turn at 200 mps ▪ Secondary flies straight-line path at 200 mps ▪ Swerling case 1 fluctuations with drop outs ▪ SNR ranges 7-12 dB • Radar operations ▪ Two reports per scan (max) ▪ No false alarms in this case ▪ 10 second update rate
	Reactive strategy	• Same 4-state promotion logic as used in Case 1 • No maneuver noise applied • Level 1 detections
	Proactive strategy #1	• Use 4-state promotion logic on mid priority (MP) primary track • Level 3 detections for high priority (HP) target
	Proactive strategy #2	• Use 5-state promotion logic HP target • Apply 0.9 g. of maneuver noise to primary track • Apply TBD logic to dropped tracks which occur in a designated HP region • Level 3 detection
4	Track coasting through large shadow zone	• Two targets flying straight line paths enter shadow region ▪ Both tracks demoted normally and eventually dropped ▪ HP track reports stored in buffer • Radar Operation ▪ Swerling case 1 fluctuations with drop outs ▪ Two reports per scan (max) ▪ No false alarms in this case ▪ 10 second update rate Note: shadow region possibly due to jamming, LOS blockage, severe clutter, ground traffic, tangential velocity, etc.

Table 13: Tracker Demonstration



- **LEVEL 3 DETECTION REPORTS**
- **ASTERISK DENOTES PRIMARY TARGET**
- **CIRCLE DENOTES SECONDARY TARGET**

Figure 15: Tracking Example Case 3 Detection Data

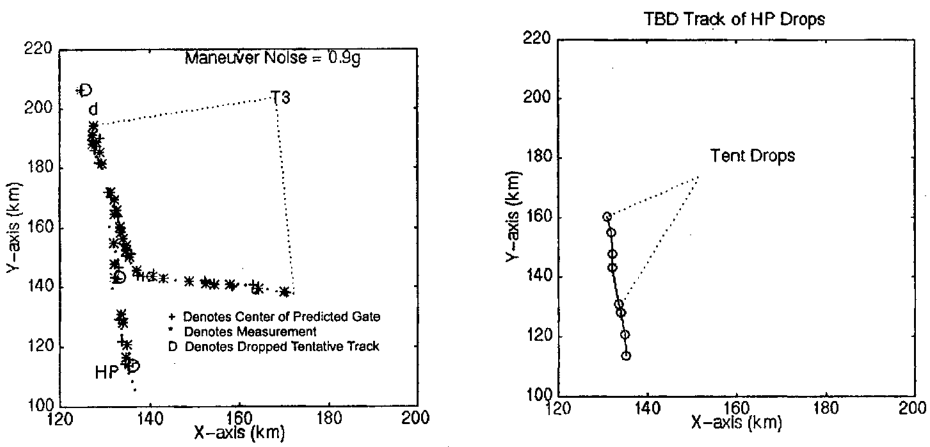


Figure 16: Proactive Tracker Output (Strategy #2)

4.3.3.5 Gate Scenarios

Three types of association gate scenarios were provided in the KBT software, including tracking rules associated with their application. The first type uses a single elliptical gate centered on the predicted point and oriented along range/cross-range. Its size depends on both measurement error and selectively injected maneuver noise that is added to the prediction covariance matrix. Association occurs when the measurement falls within the gate. The second gating scenario uses two gates: a measurement gate and a maneuver gate, both centered on the predicted target position. The measurement gate is oriented along range/cross-range and has a size dependent only on the measurement error. The maneuver gate is oriented along the track/cross-track and its size depends only on the prediction covariance matrix. Association for this case occurs when the measurement falls within either gate. Finally, the third gating method is similar to the second in that two range and track oriented elliptical gates are used and the measurement gate size depends only on the measurement error. However, the measurement gate is centered on the measured data point and the maneuver gate is centered on the predicted position. Also, the maneuver gate size is determined by assumed worst-case target kinematics. For this case, association occurs when the measurement ellipse intersects the maneuver ellipse.

4.3.3.6 Knowledge-Based Tracker Rule Book

Volume IV documents the knowledge-based tracker work for KBSTAP [10]. The "Knowledge-Based Tracker (KBT) Rule Book" is provided as Volume V [11].

The main body of Volume IV documents a basic multi-target knowledge-Based Tracker capability imbedded in its own GUI. This system was developed semi-independently of other KBSTAP efforts to demonstrate and prove the KBT concepts as discussed above. The basic capability allows the use of the following three types of tracking filters:

- An uncoupled two state $\alpha\beta$ filter with position and velocity component states
- An uncoupled three state Kalman filter with position, velocity and acceleration component states
- An extended four state Kalman filter with both x and y position and velocity component states

The report discusses the high level flow of the tracker software, and four knowledge based rules that were developed and tested as part of the basic capability. These are:

- A Maneuver/Obstacle rule that specifies the use of shaped elliptical gates
- A Maneuver/Obstacle rule that specifies the use of a gate whose shape is determined solely by centripetal turning mechanics
- A Shadow rule that provides a means of preserving firm tracks that enter regions shadowed from the radar line of site
- A Discrete rule that allows the tracker to coast through any region containing a large discrete tagged by the KBSTAP processor and to essentially ignore it

The bulk of Volume IV is contained in Section 3.0, which documents the basic tracker software, which was written in Matlab 5.1. Forty-nine modules are documented by:

- Module name
- Calling module
- Called modules
- Inputs
- Outputs
- Globals
- Module description

The separate KBT Rule Book (Volume V) is itself comprehensive and is divided into three principal subsections. The initial principal subsection is conceptual in nature providing discussions of the proactive KBT tracker design rational and assumptions, and separate discussions of issues concerning a tracking figure of merit, impacts of surveillance mission requirements, and control of target priority. The second subsection provides conceptual descriptions of the database, post-detection processor, other KBT components, and an operational concept and functional flow. The third subsection presents 25 rules, which the main report suggests are potential rules. Each rule is stated in boldface type. A discussion of that rule describing the rationale for the rule and impact on the overall KB system then follows. Finally, the interface requirements among the KBT, KBC and the radar systems are provided.

4.3.4 Detection Processor

The current approach to the KBSTAP detection process was presented at the 06 May 1997 TIM. Segmentation would be performed based on known findings (Baldygo, [2]) that significant benefits can be expected from segmenting the range-angle/Doppler data into fairly homogeneous regions. KBSTAP would use 2-D segmentation based on existing Expert System CFAR criteria and augment this with knowledge from mapping data. After segmentation, CFAR rules would be based on the known environment. Bases for CFAR selection are given in Table 14.

Environment	CFAR Algorithm
Homogeneous	Cell Averaging
Clutter Edges	Greatest-Of
Interfering Targets	Ordered Statistics
Clutter Edges + Interfering Targets	Trimmed Mean

Table 14: CFAR Selection Rules

After knowledge based filtering and CFAR segmentation, data could be expected to be largely homogeneous and in most cases call averaging would be sufficient. Figure 17 illustrates the CFAR logic and the proposed detection process.

The non-homogeneity detector was implemented and applied to the MCARM data (Acquisition 575/Flight 5). As a baseline example, the training sample data was selected about the GIP test statistic mean. The sample window was then biased towards larger values (clutter and targets) and lower values (noise) to emulate knowledge-based control of sample selection. This investigation confirmed that knowledge-based control of the non-homogeneity detector would be effective.

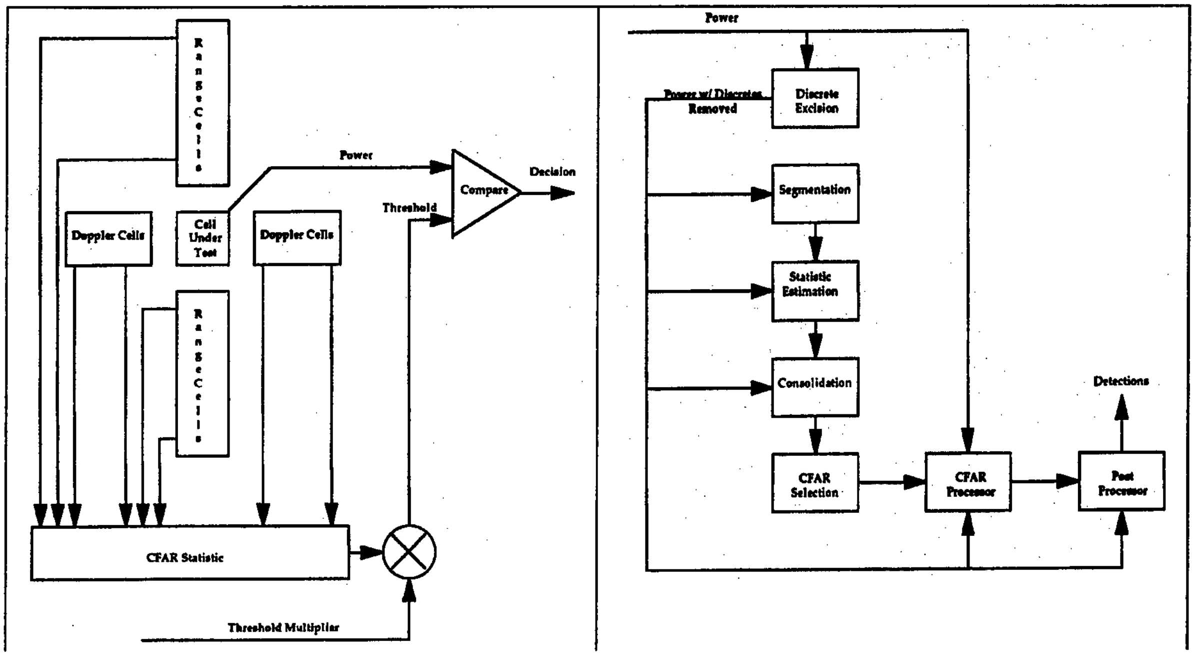


Figure 17: KBSTAP Detection Process

4.4 STAP University Initiatives

The implementation of space-time adaptive processing does not come with simple solutions. Although the equations that govern the design, behavior, and motion of the adaptive process are well understood and sound, even for systems that have a large number of adaptive weights (DOF), application in a near real-time environment has proved challenging. When attempts are made to implement such algorithms, the massive amount of data that must be processed becomes enormous. This complexity can easily overcome even the most advanced of computer architectures. Therefore, methods for efficiently handling of the data and its associated operations needed to be devised when adaptive algorithms are considered.

In 1995 DARPA and AFRL launched a Space-Time Adaptive Processing University Initiative (STAP UI) program in an attempt to manage these difficulties. The stated purpose of this program was to foster advances, through the academic community in developing state-of-the-art algorithms and processing methods that would be feasible with state of the art computer architectures and would enhance the performance of adaptive radar systems, particularly the airborne type. Many investigators participating in the STAP UI accomplished worthwhile results. They approached the problem by matrix rank reduction techniques, developing a variety of schemes constraining the number of DOF to a minimum, while maintaining the operation of the radar system at an acceptable level of performance. So that the KBSTAP program could take advantage of this research by applying it directly or extending it, Decision Sciences Applications, Inc. (DSA) was asked to conduct a study and report on potentially valuable outcomes of the STAP UI program [9]. Table 15 summarizes the university programs investigated and their principal foci of investigation as stated in the DSA report. The reader is directed to the reference, which provides the theoretical details and equations on which their work is based, assessments of the significance of their work, and relevant references.

University of Colorado
• Adaptive and non-adaptive processing applied to the generalized sidelobe canceller with the use of non-uniformly spaced weights (in space and time). The non-adaptive portion reduces the required adaptive DOF.
• An iterative steering vector estimation approach that uses received signals of opportunity and should eventually lead to ideas for data based self-calibration.
• Application of linear constraints to STAP processing and detection to significantly reduce the number of adaptive DOF without loss of processor performance.
• RADAR signal processing in the areas of:
▪ STAP modes and detectors
▪ Covariance estimation
▪ Hot clutter mitigation
Cornell University
• Performance evaluation of benchmark implementations in different architectures
• Real world performance using MCARM data
Georgia Institute of Technology
• Extended the generalized sidelobe canceller through the application of rank reduction algorithms.
• Modeled spherically invariant random vectors (SIRV) from alpha stable exited processes in their work to treat non-Gaussian clutter cases.
Northeastern University
• Polarimetric STAP. The polarized senses, vertical vs. horizontal, (or any two orthogonal senses) have adaptive weights applied to them in addition to aperture and Doppler weights
• Adaptive canceller architectures to handle terrain scattered jamming and mapped their polarization algorithms into a parallel-processing computer
• A general vector subspace formulation for multidimensional adaptive detection algorithms. Folded into this framework was the use of SIRVs to model the data.
• Theoretical development of their generalized Adaptive Matched Filter (AMF) receiver for non-Gaussian interference.
• Data-generated metrics for meaningful comparisons of different STAP algorithms
Polytechnic University of Brooklyn
• A direction finding method based on a high-resolution joint Doppler-azimuth estimation with clutter modeling as a knowledge base for initializing the STAP process
▪ Used a tiled approach for the implementation of the algorithms to massively parallel architectures
• Block Augmented Matrix (BAM) inversion techniques for reduced DOF STAP that minimizes computational burden, Predictive-Transform (PT) STAP, and parallel architecture implementation.
Syracuse University
• A direct least mean squares approach for the adaptive process. By slightly oversampling the noise field, weights were generated after one snapshot, enabling operation against repeater jammers as well as broad band cw jammers
• Addressed the problem of array element mutual coupling
• In the architecture area , evaluated the relationship between data partitioning, number of matrix rows and columns, number of nodes, and timing
University of Southern California (USC)
• Airborne surveillance radar requirements under four topics
▪ Rank reduction methods using the cross-spectral method
▪ Derivations of performance limits for other rank reduction methods
▪ Higher-order statistical signal processing
▪ Parallel mapping of signal processing algorithms onto high performance computing platforms.

Table 15: Summary of STAP University Initiatives

The STAP/UI Investigators addressed a wide variety of topics. Major areas included rank reduction of otherwise large degree of freedom adaptive systems, beamspace cancellers, imbedded CFAR, mitigation methods for terrain scattered interference, angle/Doppler estimation, detection in non-Gaussian noise, and implementation of adaptive algorithms into computer architectures. In all these cases, the investigators used simulated data as well as actual data to validate the performance of their algorithms. For example, in certain scenarios, rank reduction methods provided nearly equal performance when compared to the fully adaptive system, allowing great savings in the commitment of computational resources. These are situations in which the entropy of the noise environment lies within the bounds of the reduced rank adaptive process capability. The innovative aspect of the rank reduction approach is efficient allocation of those remaining adaptive resources so that the attendant nulling is focussed primarily on the interference sources.

The cited report [9] reviews the achievements of each university program and the applicable equations. To take advantage of this work, it is recommended that a comparative evaluation of the performance of the developed algorithms be conducted. The evaluation would make use of simulated and experimental data to determine the relative enhancements provided by the algorithms. Measures of performance would need to include signal to noise ratio, error and false alarm rates, and detection probability as affected by the various algorithms. Degree of complexity and demands on computational resources would also have to be considered.

4.5 KBSTAP Integration

During the course of the KBSTAP program a substantial effort was invested in the establishment of the theoretical and realizable grounds for constructing an integrated KBSTAP system. Of practical necessity, the individual components, particularly the filtering and tracking processes, were to a large degree investigated independently. In each case, however, the idea that there would be a repository of affiliated knowledge available to be applied in optimizing performance decisions remained the guiding principle.

The investigation of knowledge-based filtering and knowledge-based tracking did result in the implementation of demonstration systems where testing and experimentation led to the effective algorithms and rules for applying them. These rules enabled performance to be tuned to the environmental conditions at hand. Assembly of these demonstration systems resulted in a substantial software development, which was accomplished with consideration of modularity for eventual integration and expansion. The next step was to pull this work together into an integrated system, which would afford an opportunity to demonstrate a coordinated cycle of knowledge based radar processing and could be easily enhanced when new developments warranted it. Integration and testing of a complete laboratory KBSTAP would also be a necessary step leading to an eventual field demonstration phase. For the most part, the software developed to demonstrate knowledge-based filtering and tracking could be reused in the integrated KBSTAP.

Figure 18 is block diagram of the KBSTAP system that would serve as basis for KBSTAP integration. This design evolved from the initial concept as presented in the DP [3] and the development and experimentation that took place as the KBSTAP program progressed. The four major centers of processing, namely control, filtering, detection and tracking, are shown underlying lower level elements of the diagram. The Graphical User Interface (GUI) is the subsystem from which radar specialists can exercise control of the system. Through this subsystem the user can navigate through any part of the system. Working within an X-Windows environment the user can display a mapping of the operational theater that depicts a topographical view including the platform and targets (if any), and performance characteristics of various KBSTAP elements. In addition to the map view, the GUI provides menu buttons from which to control the functionality of the system. Figure 19 is an example of a GUI display simultaneously showing a terrain map, a tracker report, and range profile from the filter module. All user commands generated through the GUI are acted on by the KBC.

The KBC receives the requests from the user, through the GUI, and then parses them into control commands for the affected subsystems adding knowledge domain information as required. In addition to processing requests from the GUI, the KBC also receives feedback from the tracker process, which is acted upon by passing revised information for maintaining optimum system performance to the filter process. The filter process receives a data cube resulting from the current CPI. From the information contained in the data cube and the intelligence passed from the KBC, the filter process will select non-adaptive or adaptive algorithms to scrub the data. Application of pre-determined rules that consider the current radar parameters and theater conditions leads to the best course of action.

The detection process receives best possible (filtered) range/Doppler data and generates detection reports that are subsequently passed on to the tracker process. The tracker process uses the detection reports in combination with environmental data and pre-defined rules to maintain multiple target tracks in regions of obstructions and interference. Tracking reports are generated and fed back to the KBC, which can then take steps to enhance overall system performance for the next processing cycle.

Development and testing of the filtering and tracking demonstrations resulted in the availability of a great deal of reusable software. To make this software work within an integrated system as depicted in Figure 18, data items that needed to be interchanged between major processes, or would be shared by processes, needed to be identified and databases established. An initial listing of these data elements is provided in a separate document. The listing is organized according to the four principal processes of control, filtering, detection, and tracking.

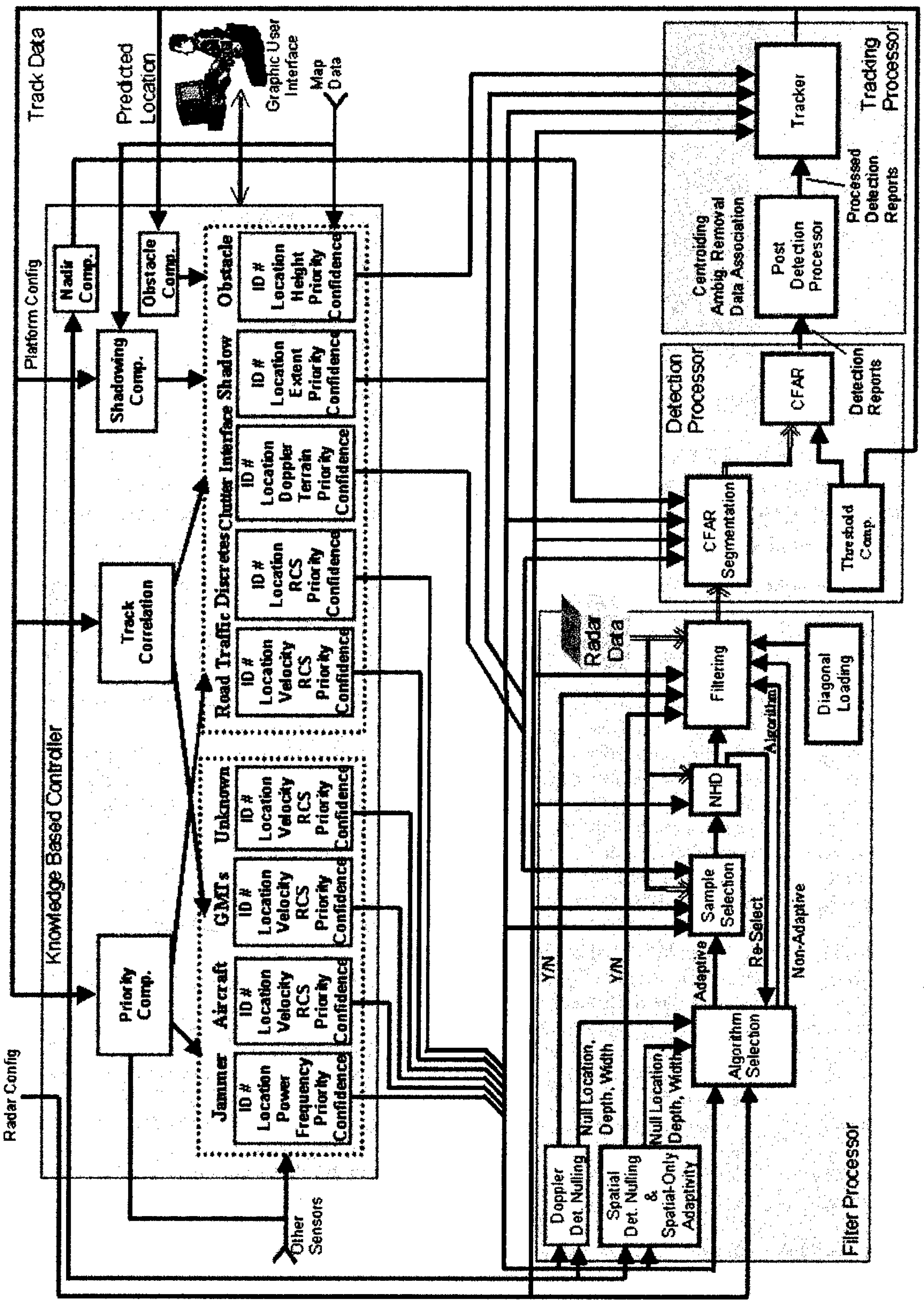


Figure 18: KBSTAP Block Diagram

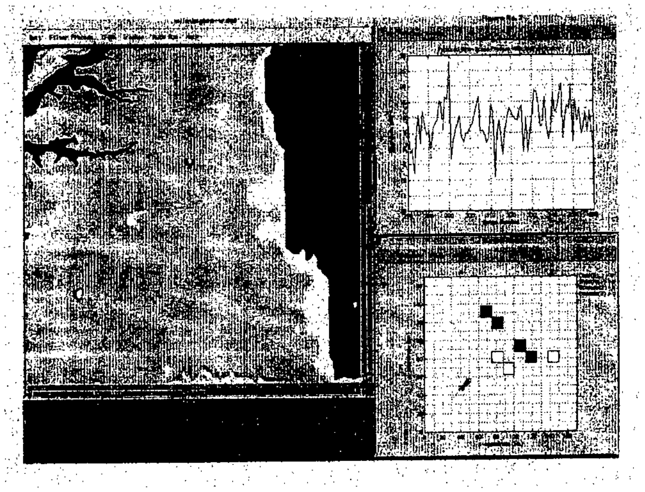


Figure 19: KBSTAP Graphical User Interface

A first-level design of KBSTAP was based on five major processes and four system states that were seen to require unique considerations. A fifth process was separated from the KBC process to perform overall KBSTAP performance monitoring. Table 16 summarizes the functions performed within the 20 process-state combinations. The following numbered items elaborate on these functions.

KBSTAP Process/State Design

State	Process	Function
Pre-Flight	**Performance**	**1)** Locate and load all Discretes, Clutter Boundaries, Shadow Regions Potential Jammers, Obstacles - Set System Parameters
	Control	**2)** Locate and load all Discretes, Clutter Boundaries, Shadow Regions, Potential Jammers, Obstacles - Set System Parameters
	Filtering	**3)** Define initial settings and performance measure thresholds
	Detection	**4)** Define initial settings and Thresholds for Pfa
	Tracking	**5)** Locate all Discretes, Clutter Boundaries Shadow Regions, Potential Jammers, Obstacles - Define initial settings and performance measure thresholds
Initiate System & Initial Transient States (4 to10 CPIs)	**Performance**	**6)** Monitor System
	Control	**7)** Initiate System and Monitor
	Filtering	**8)** Execute Non- STAP Algorithm. - Compute No. of Secondary Rings - Run NHD - Compute Beam Performance, Determine Null Weights - Determine STAP feasibility
	Detection	**9)** Compute and Adjust Thresholds for Pfa
	Tracking	**10)** Initiate Tracks - Compute Performance Measures (Number of Correct Tracks, Number of Dropped Tracks, Number of Incorrect Tracks)
Correlation, Assessment, Learning (1 to 2 Complete Scans of a Defined Scene/Area)	**Performance**	**11)** Correlate Discretes, Clutter Boundaries, Shadow Regions, Potential Jammers, Obstacles - Evolve Rules - Insert Synthetic Targets Measure Performances
	Control	**12)** Correlate Discretes, Clutter Boundaries, Shadow Regions, Potential Jammers, Obstacles Evolve Rules
	Filtering	**13)** Compute Nos of Sec. Rings, Run NHD, Compute Beam Performance Measures, Set Nulls, Determine When and Where STAP is Feasible - Evolve Rules
	Detection	**14)** Compute Detections – Re-compute and Adjust Pfa Thresholds - Evolve Rules
	Tracking	**15)** Correlate IFF Data with Tracks - Compute Performance Measures - Number of Tracks, Number of Dropped Tracks, Number of Incorrect Tracks - Evolve Rules
Steady State	**Performance**	**16)** Insert Synthetic Targets- Measure Performances - Change Rule Sets Accordingly
	Control	**17)** Measure Performances Change Rule Set Accordingly
	Filtering	**18)** Measure Performances - Change Rule Set Accordingly
	Detection	**19)** Measure Performances - Change Rule Set Accordingly
	Tracking	**20)** Measure Performances - Change Rule Set Accordingly

Table 16: High Level KBSTAP Process-State Design

4.5.1 & 4.5.2 *1-2 Pre-Flight for KB Processes.*

The hypothesized location of discretes, clutter boundaries, shadow regions, potential jammers, and obstacles need to be loaded into the system. This can be performed in at least 2 ways.

1. Load the location of all these entities into one table and as the system progresses to the learning state it will find entities that it will check against the list. As they are verified, their status will be changed from hypothesized to identified and their parameters updated accordingly. As new entities are found they will be entered into the table also as hypothesized and as they are verified, with data obtained from more than one track they will be upgraded to identified.

2. Load all that are hypothesized into separate tables based upon their type (i.e. discretes, obstacles, shadow regions, aircraft, etc.) and as they are confirmed they will be marked identified. As new entities are' found, they can be placed into a general table as in 1. and, as they are confirmed, they can be moved to their proper table.

The classification and storage of the different entities can be done in the following manner.

Road traffic (road ID #, LL1, LL2, LL3, . . ., LLn, priority, confidence (= 0 when first loaded, i.e. hypothesized))

LLi implies latitude and longitude of points on the earth that defines a straight line approximation of the road.

Discretes (discrete ID #, LL1, LL2, priority, confidence,)

Clutter types (clutter type ID #, LL1, LL2, LL3, . . ., LLn, priority, confidence (= 0 when first loaded))

The location points define the boundaries for the clutter type such as urban, forest, ocean, etc. It is assumed that every point within the boundary is the same type of clutter. This will help the homogeneous clutter model to choose secondary data before NHD is applied.

For shadow or obstacle type (shadow/obstacle ID #, LL1, LL2, LL3, . . ., LLn, maximum height, priority, confidence (= 0 when first loaded),)

The Location points describe the base of the shadow or object.

In addition, if it is known from intelligence sources where jammers are located, they may be entered in a fashion similar to discretes but with a zero confidence.

4.5.3 *3 Pre-Flight Intelligent Filter Process*

These data represent antenna characteristics that will not change during flight, e.g. number of antenna elements and their configuration, antenna tilt angle and pointing direction, and location of antenna on the A/C. It will also contain data that will be required for its first processing sequence. These data may change during flight, e.g. PRF, transmitter frequency, size of the data cube, and bandwidth of signal.

Performance thresholds must also be set for evaluating antenna beam distortion.

4.5.4 *4 Pre-Flight Intelligent Detection Process*

These data represent data that are initialized but are not necessarily fixed, e.g. range resolution, Doppler resolution, top percentile for TM-CFAR, bottom percentile for TM-CFAR, and beam time. Data related to performance measures are also set such as design probability of false alarm for normal, low and very low thresholds.

4.5.5 *5 Pre-Flight Intelligent Tracking Process*

This state has data requirements similar to the pre-flight KB processes states. This state processing is done once, and all three processes would have access to the data with their respective rules. This state also requires setting of the tracking process performance measures, e.g. number of correct tracks and number of dropped or lost tracks.

4.5.6 *6 KB Performance Process and Initiate System and Transient State.*

For this initial state, the process will monitor the system queues for a number of potential targets and registration of obstacles, discretes, clutter boundaries, shadow regions, and jammers.

4.5.7 *7 KB Control and Initiate System and Transient State.*

For this initial state the process will initiate the antenna processing and monitor the system queues for data cube compatibility, auxiliary data correlations, feedback from the different processes, system errors, number of potential targets and registration of obstacles, discretes, clutter boundaries, shadow regions, and jammers.

4.5.8 *8 Intelligent Filter Process and Initiate System and Transient State.*

Execute non-STAP algorithms, determine the number of secondary rings for each cell under test given the stored terrain features, run the non-homogeneous detector algorithm if necessary, compute beam performance, determine null weights based upon assigned known interferers to null. Note for this state, nulling is probably not done but enough data should be gathered to perform correlations, so that KB control can determine if nulls should be placed on certain interferers and not others. Determine STAP feasibility.

4.5.9 *9 Intelligent Detection Process and Initiate System and Transient State.*

For these initial states, the process will implement thresholds as assigned, will default to the standard detection cell averaging algorithm, and use standard window sizes unless it is known to be near boundaries.

4.5.10 *10 Intelligent Tracking Process and Initiate System Transient State.*

Multiple CPIs are required to initiate tracks. Correlations with objects and shadow regions are begun and the system will begin to compute performance measures (number of correct tracks, number of dropped tracks, number of incorrect tracks) and sorting of tracks. It will also report back tracks and potential correlations.

4.5.11 *11 KB Performance Process and Correlation, Assessment, and Learning State.*

This process will make use of the correlations performed by KB control (in state 12) for the first portion of the processing. That is, until it is satisfied that it has correlated most of the discretes, clutter boundaries, road traffic, and shadow regions. Once these have been confirmed the process will insert synthetic targets of varying sizes and velocities to test the performance of the total system. During the first and second complete scan of an area the KB performance process will be able to determine if the performance measures have improved. Based upon these results, the performance process may place targets in other locations and/or direct the controller as to where it should not use STAP.

4.5.12 *12 KB Control and Correlation, Assessment, and Learning State.*

There are two levels of correlation that are required: (1) position of the above entities and their location within a defined range ring and (2) the power level at the receiver given the distance between the antenna and the entity. Note the definition of the range rings relative to the earth encompass different entities as the aircraft moves. In addition, as the aircraft moves to or away from entities that require nulling, the system may or may not want to place a null in their direction. Correlating entities by power may be done as discussed next.

Road traffic power (road ID #, peak power divided by average peak power over a defined window for CPI #) - correlations will be done by the <u>road object correlator</u> along with the data from the detector and tracker. Power can be used to determine if the return signal varies wildly from falling off at approximately $1/R^4$ (when corrected for the speed of the aircraft) at the projected location and if the majority of targets/tracks that originate from the location follow a road traffic pattern.

Discrete power (discrete ID #, peak power divided by average peak power over a defined window for CPI #) - Correlations will be done by the <u>discrete object correlator</u> along with the data from the detector and tracker indicating that it is not moving. Verify that the power falls off as $1/R^4$ (when corrected for the speed of the aircraft) and the objects do not move, i.e. do not generate a track.

Shadow/obstacle power (shadow/obstacle ID #, peak power divided by average peak power over a defined window for each CPI #) - Correlations will be done by the <u>shadow/obstacle object correlator</u> along with the data from the detector and tracker indicating that it is not moving and it is indeed an area that provides a shadow or an obstacle that low flying objects must fly around. Correlations with minimum power responses at range and angles where potentially shadowed areas can be computed and compared. Remember that the shadowed regions loaded at pre-flight are computed from USGS, or NIMA databases, with an assumed flight path. If the databases are old then the terrain may have changed over time.

As data is obtained from IFF responses, outside sources, and other sensors, jammer objects, aircraft, ground moving targets, and all unknown objects will be able to be updated. Numerous data sources will be needed in order to register each CPI with ground "truth".

4.5.13 *13 Intelligent Filter Process and the Correlation, Assessment, and Learning Stage.*

In this state, rules need to be built as to when STAP is feasible and can be applied or conversely when STAP is not feasible and should not be used. It is assumed that the radar is flying in a known pattern and will be looking at the same scene or area each time it flies the same pattern. During the first complete scan of the defined area the system should execute a standard non-STAP algorithm. The KB performance process should place targets in non-homogeneous areas, e.g., near roads and clutter boundaries. KB Control does not know the position and type of synthetic targets placed. In the second complete scan Controller should attempt to use STAP wherever it can be given the required number of samples available and required.

A rule must consider when there exists a sufficient number of training range rings that are of the same type to perform STAP. A method for doing this is to correlate each range ring number with the terrain map to identify where there are discontinuities, major roads, etc., and each region or sector-range must be labeled with a terrain type that is assumed similar. These labels will change as the aircraft flies different paths. The classification code range ring correlator in collaboration with the intelligent filter process will do this correlation procedure. A method is to use the major classification codes used in the USGS database, e.g., urban, forest, water, etc. These partitions can be evaluated for their homogeneity by using NHD. With a combination of the pre-flight loaded database, the use of the radar returns and the NHD, the system can "learn" which areas are homogeneous and evolve its rules as to which filter algorithm to use.

During this state the controller will assign a low, medium, and high performance threshold levels for beam performance. These will be recorded for each instance the controller requires the intelligent filter to place nulls in certain locations dictated by a known or hypothesized discrete or jammer. The results will be passed on to KB control to be evaluated with the feedback from the detection and tracking processes. Based upon these results, KB control may assign the filter process different nulling levels for the interferer.

4.5.14 *14 Intelligent Detection Process and the Correlation, Assessment, and Learning Stage.*

For this state, detection will use the correlation data provided by KB control to determine where boundaries are located. For those test cells within homogeneous regions, the standard detection cell averaging algorithm and standard window sizes will be used. For those test cells near boundaries, the rules within the CFAR processes will choose reference cells, algorithms, and window sizes as developed under the ES-CFAR program. The process will perform detections, implement thresholds as assigned, re-compute and adjust Pfa thresholds, evolve rules to apply the standard cell averaging rules, when to apply more sophisticated algorithms, and when to recommend changing detection thresholds.

4.5.15 *15 Intelligent Tracking Process and the Correlation, Assessment, and Learning Stage.*

Objects, FAA data, and shadow regions will be obtained from KB control and correlated with tracks. Correlations of dropped tracks and highways will be performed with KB control.

Performance measures (number of correct tracks, number of dropped tracks, number of incorrect tracks) and sorting of tracks will be computed. The tracking process will also report back tracks and correlations reported by KB control that are not supported by the tracking. These will have to be settled by KB control and the other processes. As corrections are made the system will evolve its rules.

4.5.16 *16 KB Performance Process and the Steady State.*

The performance process will constantly measure the performance of all other processes to determine whether the system is performing better or worse than its previous processes. The process will continually look for changes or requests submitted by the user or changes in data from outside sources. It will monitor performance by checking the Beam Pattern Performance data, Detection data, and Track data. It will insert known RCS synthetic targets at locations where there are boundaries in terrain types and evaluate the detection capability of the system. By placing different targets at different locations it can determine if the current rules are performing well. If they are not, the rules being used by KB control will be changed.

4.5.17 *17 KB Control and the Steady State.*

KB control will also have access to the same performance measures as discussed in 16. Based upon these performance values it will interpret its current rules and process data accordingly. The rules KB control can change are based upon a process's reported data and user request changes, such as change in direction or flight path, detection levels for controlling false alarms if there are too many tracks, etc.

4.5.18 *18 Intelligent Filter Process and the Steady State.*

This process will monitor its own performance related to beam pattern and will change its rules based upon such things as notching of too many jammers or discretes thereby using too many DOF, change in direction of the aircraft thereby changing the degree of notching required per jammer and discrete, etc.

4.5.19 *19 Intelligent Detection Process and the Steady State.*

This process will measure performance based upon number of detections and number of false alarms, and increase or decrease the threshold level, change window sizes for CFAR algorithms, and change rules for choosing CFAR algorithms based upon previous flights over the same or similar clutter interfaces.

4.5.20 *20 Intelligent Tracking Process and the Steady State.*

This process will measure performance based upon number of correct tracks, missed tracks, and number of false tracks. Based upon these numbers and the terrain the process will adjust its rules and thresholds to increase its performance.

5.0 Summary of Operational Status

The integrated concept of KBSTAP that was published in the Design Plan [3] is implemented to the extent that it is a functioning laboratory system with at least basic level of functionality for all planned features. It provides a high level of user accessibility with multiple modes of display that makes it an effective vehicle for demonstration of knowledge-based radar principles, continued development and testing of new radar technology, and the training of radar engineers. The knowledge-based processes for filtering, detection and tracking that were developed independently during the course of the project have been integrated under the control of a Knowledge Base Controller (KBC). Refer to Figure 18.

The KBC as implemented enables the KBSTAP concept by orchestrating the preferred application of algorithms given the current environmental conditions. The KBC controls user access through an X-Windows graphical user interface and maintains a knowledge domain database populated with feature data on the local operating theater. The KBC incorporates software to dynamically update the knowledge domain database to indicate object identifications and level of confidence and extract map features. Feature extractions are used to predict shadowed areas in the target flight path, compute the likelihood of a target maneuver to avoid known obstacles, and correlate target paths with road maps. Figure 18 indicates the range of feature data that is available in the database.

The KBSTAP filtering processor was implemented by adapting the software algorithms developed for the filtering demonstration capability used to investigate the most effective use pre-STAP and STAP filtering. A baseline of macro and micro rules for applying these algorithms, developed from extensive experimentation with MCARM data, are implemented. The conditions and which filtering algorithms should be applied are summarized in Table 17.

Filter Processor Rules	
Condition	**Rule**
- Macro Rules -	
Default STAP Algorithm	Factored STAP
Insufficient Homogeneous Samples	Non-Adaptive Approach DPCA, MTI
Slow Moving Target (Default Algorithm)	Factored STAP
High Speed Target (Default Algorithm)	Beam-Space Post Doppler ADPCA
- Micro Rules -	
Homogeneous Terrain	Sliding Widow
Non-Homogeneous Terrain	Non-Homogeneity Detection
Land/Sea Interface	Segmented NHD
Potential Traffic Interface	Mapping Data + NHD
Discretes	Mapping Data

Table 17: Filtering Rules

In a similar manner the detection processor was implemented using several CFAR algorithms and associated rules derived from experimentation with MCARM data. Table 18 lists these conditions and which algorithms to apply.

Detection Processor Rules	
Condition	**Rule**
Default Algorithm	Cell Averaging
Homogeneous Terrain	Cell Averaging with Sliding Window
Non-Homogeneous Terrain	Cell Averaging with Reference Cells Taken from the Output of NHD
Land/Sea Interface	Cell Averaging with Segmented NHD

Table 18: Detection Rules

A multi-target, proactive tracking processor was also implemented by adapting software from a demonstration facility. The following filtering capabilities are available to the processor:

- Two-state $\alpha\beta$ filter with position and velocity states
- Three-state Kalman filter with position, velocity, and acceleration states
- Extended four-state Kalman filter with both x, y, and Doppler states

Use of these algorithms are dictated by the conditions and rules Table 19.

Tracking Processor Rules	
Condition	**Rule**
Default Track Filter	3-State Kalman
Obstacle in the Flight Path	Expected Maneuver. Increase gate size or use "Smile Gate"
Shadow	Keep a Straight Line Coasting for Firm Tracks
Discretes	If a Discrete Falls in a Track Gate, the Track is Treated as Shadowed

Table 19: Tracking Rules

6.0 Future Directions

The KBSTAP effort resulted in an integrated, laboratory system that demonstrates many advantages of knowledge-based radar processing. As is often the case in such a project there are further steps to be taken to move the KBSTAP concept toward an operational configuration. Also, as research and development unfolds in such an effort, many areas of technical refinement are suggested. This section makes several recommendations, not necessarily all inclusive, as to where future effort toward an operational KBSTAP should be focused.

6.1 A Proven Framework is in Place

The KBSTAP program resulted in several important breakthroughs. Of greatest importance is a framework (Figure 18) that takes into account the complex interrelations between filtering, detection and tracking, and orchestrates these links for optimum performance through exploitation of topographic and other knowledge about the immediate operating theater. This architecture did not drop out of a clear sky but evolved from a background of previous research, and the steps taken to implement and demonstrate knowledge-based selection of algorithm regimes. These efforts worked to separate out practically doable strategies from the purely theoretical and resulted in associated rules about the best application of antenna configurations, pre-adaptive and adaptive filtering, detection thresholds, and proactive multi-tracking in the presence obstacles and other interference. This understanding is reflected by the architecture.

From the beginning of the effort it was understood that, by its nature, a knowledge-based view of radar processing offered many possibilities for extendibility and dual use. The KBSTAP framework as it now stands provides for the incorporation of new or different knowledge, rules, algorithms, or applications without a ground-up reinvention. Therefore the framework offers opportunities for extension of knowledge-based radar processing into even more efficient airborne military radar systems, and other military platforms, such as ships at sea. There are also opportunities for knowledge-based radar to be migrated to traffic control, weather, ground imaging, and other radar systems, both within the military and in related commercial sectors. Ideally only the relevant parameters and applied rules would need to be modified. It is recommended that opportunities for dual use applications and funding be investigated.

6.2 Expansion to Bistatics

It was intended that KBSTAP would be developed for the bistatic configuration but, because of the demands of many theoretical issues concerning efficient Doppler processing and the physical characteristics of interference, the implications of bistatic radar were not fully explored. Now that a monostatic KBSTAP architecture has been stabilized and many algorithmic issues have been resolved, it is recommended that the KBSTAP algorithms and possibly rules be generalized to the bistatic case. While the bistatic case is mainly a geometric problem, issues related to dynamic changes in the angle and distances between antenna and receiver may require extensions of existing algorithms and may affect rules that use thresholds for making decisions that affect antenna, filtering, detection, and tracking. Generalization to the bistatic case will also improve KBSTAP's adaptability to new applications.

6.3 Airborne Test Bed

The work performed during this effort demonstrated the effectiveness and feasibility of intelligent Doppler radar processing, but the experimentation and testing as accomplished was limited to a single set of collected flight data and injected targets. KBSTAP should now be migrated to a test bed situation where it would be connected to an airborne radar platform and could evaluate the performance of existing algorithms and knowledge-based rules under actual field conditions. It is recommended, particularly for the bistatic case, that a KBSTAP test bed be connected into a receiver so that the complete transmit and receive cycle of knowledge-based radar can be tested dynamically.

6.4 Super-Computer Architectures

To be effective in an operational environment KBSTAP will have to perform in a near real-time fashion. Doppler radar processing is high throughput enterprise characterized by multi-channel signal processing, and is an obvious application for parallel processing. KBSTAP should be investigated for parallelization opportunities both on a functional level and on an algorithmic level. On a functional level KBSTAP may be divided and conquered such that functions like filtering and tracking (or probably even finer partitions) operate in parallel, and semi-independently, on possibly physically separated processors. This would, however, demand high-bandwidth, inter-processor communications and an overarching logic for synchronization. Consideration of bistatic radar would seem to imply some level of functional partitioning and inter-processor communications to coordinate antenna arrays and receivers separated by significant distances.

At the algorithmic level, parallelization may be used to perform simultaneous calculations on array elements as required by the algorithm. A great deal of algorithm development such as that found in the report on the STAP UI [9] was devoted toward reduction of the computational burden required by multi-array signal processing. Parallization may be applied to speed up these already efficient algorithms or to mitigate the compromises that were made only to reduce computational burden. This fine-grained kind of parallelization, however, requires tightly coupled microprocessors as most often deployed within supercomputer architectures, which may not necessarily be field deployable.

AFRL's new SKYComputers supercomputer may serve to provide a vehicle for both functional and fine-grained problem partitioning. This machine as built is a parallel microprocessor architecture but is also ruggedized for field deployment, and modularized for reconfiguration into one or more computers. For example, one module may be airborne to control beam generation while other modules may perform knowledge-based processing in a ground station. This would, of course, imply very high speed communications between air and ground.

6.5 Knowledge Based Rules

SRC suggested additional data and experience could be applied to update their initial filtering rules to a more "optimum" state, or to add new rules that would apply to airborne radar system variants not yet addressed. SRC emphasized that the rules to date can be applied with assurance only to:

- Monostatic radar
- STAP channels spatially arrayed in one dimension
- "Notching" in space only
- Sidelobe barrage noise jamming only
- The use of diagonal loading rather than mainlobe constraints for limiting target signal suppression in STAP

They caution that the existing rules were established with little regard to the processing resource load of the radar. For example, it was assumed that pulse compression and Doppler filtering would precede STAP since limited degree of freedom adaptivity was shown to be most effective if all deterministic filtering occurs first, but processing load is far grater for post compression/post Doppler STAP.

TSC delivered a basic KBT capability and a rulebook containing 25 "potential" knowledge based tracking rules. These rules should be revisited to validate their reasonableness and, if it appears warranted, installed into a KBSTAP test bed.

6.6 Formal Comparison of University Initiatives

The DSA report [9] on the STAP university initiatives brings together the achievements of these programs and their areas of application. DSA recommends that a comparative evaluation of the performance of the different algorithms be conducted to determine the relative enhancements provided by the developed algorithms. Measures of performance such as signal to noise ratio, error and false alarm rates, and detection probability as affected by the various algorithms would need to be considered as would degree of complexity and demands on computational resources. To achieve this level of evaluation, these algorithms would be integrated with the KBSTAP family of processing choices where they can be tested and compared with or in combination with algorithms already in place. Along with this, rules associated with their application would need to be added to the rulebook. Such rules are implied in the DSA study but not explicitly stated.

References

[1] Corbeil, A. and Hawkins L, "Knowledge-Based Tracker Assessment Program," Air Force Research Laboratory Final Technical Report, RL-TR-91-73, 1991

[2] Baldygo, W. et al., "Artificial Intelligence Applications to Constant False Alarm rate (CFAR) Processing," Proceedings, IEEE 1993 National Radar Conference, 1993

[3] "Knowledge Base Applications to Adaptive Space-Time Processing, Design Plan," Air Force Research Laboratory Contract F30602-95-C-0041, April 1996

[4] "Knowledge Based Space-Time Processing (KBSTAP) Software Implementation Plan, Version 1," Air Force Research Laboratory, April 1996

[5] "Radar Filtering Rulebook," Prepared for ITT Systems by Syracuse Research Corporation, Air Force Research Laboratory Contract F30602-95-C-0041, SRC TD 97-1466, October 1997

[6] "Airborne Radar Filtering – Space/Time Adaptive Processing," Prepared for ITT Systems by Syracuse Research Corporation, Air Force Research Laboratory Contract F30602-95-C-0041, SRC TD 97-1530, December 1997

[7] "Airborne Radar Filtering – Subarray/Sub-CPI Nulling," Prepared for ITT Systems by Syracuse Research Corporation, Air Force Research Laboratory Contract F30602-95-C-0041, SRC TD 98-0004, January 1998

[8] "Airborne Radar Filtering – Array Notching and Spatial Only Adaptivity," Prepared for ITT Systems by Syracuse Research Corporation, Air Force Research Laboratory Contract F30602-95-C-0041, SRC TD 98-0067, February 1998

[9] Vannicola, V., "Applicable Equations for Space Time Adaptive Processing Developed Under University Initiatives," Prepared for ITT Systems by Decision Science Applications, Inc., Air Force Research Laboratory Contract F30602-95-C-0041, DSA Report #412/1748, April 1998

[10] Morgan, C., "Knowledge Base Applications to Adaptive Space Time Processing (STP)," Prepared for ITT Systems by Technology Service Corporation, Air Force Research Laboratory Contract F30602-95-C-0041, TSC-CT109-22, February 1998

[11] Morgan, C.J. and Moyer, R.L., "Knowledge-Based Tracker (KBT) Rule Book," Prepared for ITT Systems by Technology Service Corporation, Air Force Research Laboratory Contract F30602-95-C-0041, February 1998

[12] "Some Observations on Sample Selection and Diagonal Loading for STAP," Technical Memo, February 1997

[13] Antonik, P., et al., "Knowledge-Based Space-Time Adaptive Processing," Proceedings, 1997 IEEE National Radar Conference

Acronyms/Symbols

ADPCA	Adaptive Displaced Phase Center Antenna
AFRL	Air Force Research Laboratory
AI	Artificial Intelligence
AMTI	Airborne Moving Target Indicator
CFAR	Constant False Alarm Rate
CPI	Coherent Processing Interval
DL	Detection Level
DMA	Defense Mapping Agency (now NIMA)
DOF	Degrees of Freedom
DP	Design Plan
DPCA	Displaced Phase Center Antenna
DSA	Decision-Science Applications, Inc.
GIP	Generalized Inner Product
GLRT	Generalized Likelihood Ratio Test
GUI	Graphical User Interface
IFF	Interrogation Friend of Foe
KB	Knowledge Base or Knowledge Based
KBC	Knowledge Base Controller
KBT	Knowledge-Based Tracker
KBSTAP	Knowledge-Based Space-Time Adaptive Processing
MCARM	Multi-Channel Airborne Radar Measurement
MSMI	Modified Sampled Matrix Inversion
N_{DOF}	Number of Degrees of Freedom
NHD	Non-Homogeneity Detector
N_R	Number of Reference Cells
NREP	Number of Reports for Scan
PDF	Probability Density Function
P_n	Diagonal loading
PRI	Pulse Repetition Interval
RLSTAP	Research Laboratory Space-Time Adaptive Processing
RLSTAP/ADT	Research Laboratory Space-Time Adaptive Processing/Algorithm Development Tool
RSTER	Radar Surveillance Technology Experimental Radar
SINR	Signal-to-Interference-Plus Noise Ratio
SIRP	Spherically Invariant Random Processes
SMI	Sample Matrix Inversion
SNR	Signal to Noise Ratio
SRC	Syracuse Research Corporation
STAP	Space-Time Adaptive Processing
STP	Space-Time Processing
TIM	Technical Interchange Meeting
TSC	Technology Service Corporation
USGS	United States Geological Survey
WSMR	White Sands Missile Range

☆U.S. GOVERNMENT PRINTING OFFICE: 2001-610-055-10147